

# Discontinuities in pattern inference<sup>☆</sup>

Daniel Reidenbach

*Department of Computer Science, Loughborough University, Loughborough, Leicestershire, LE11 3TU, UK*

## Abstract

This paper deals with the inferrability of classes of E-pattern languages – also referred to as extended or erasing pattern languages – from positive data in Gold’s model of identification in the limit. The first main part of the paper shows that the recently presented negative result on terminal-free E-pattern languages over binary alphabets does not hold for other alphabet sizes, so that the full class of these languages is inferrable from positive data if and only if the corresponding terminal alphabet does not consist of exactly two distinct letters. The second main part yields the insight that the positive result on terminal-free E-pattern languages over alphabets with three or four letters cannot be extended to the class of general E-pattern languages. With regard to larger alphabets, the extensibility remains open.

The proof methods developed for these main results do not directly discuss the (non-)existence of appropriate learning strategies, but they deal with structural properties of classes of E-pattern languages, and, in particular, with the problem of finding telltales for these languages. It is shown that the inferrability of classes of E-pattern languages is closely connected to some problems on the ambiguity of morphisms so that the technical contributions of the paper largely consist of combinatorial insights into morphisms in word monoids.

© 2008 Elsevier B.V. All rights reserved.

**Keywords:** Pattern languages; Inductive inference; Telltales; Combinatorics on words; Morphisms; Ambiguity

## 1. Introduction

Within the scope of the present paper, we use a *pattern* – i.e. a finite string over an infinite set  $X$  of *variables* and an arbitrary alphabet  $\Sigma$  of *terminal symbols* – as a device for the definition of a formal language. If we wish to derive a word from a pattern then we use a so-called *substitution*, which means that we replace all occurrences of variables in the pattern by arbitrary strings of terminal symbols (of course, if there are several occurrences of the same variable in the pattern then we choose the same string of terminals for each of these occurrences). Accordingly, the *pattern language* of a pattern is the set of all words that can be generated by such substitutions. Thus, as a substitution actually is nothing but a (“terminal-preserving”) morphism mapping a string in  $(\Sigma \cup X)^*$  onto a string in  $\Sigma^*$ , the

<sup>☆</sup> A major part of this paper is based on the conference contributions [D. Reidenbach, A discontinuity in pattern inference, in: Proc. 21st Annual Symposium on Theoretical Aspects of Computer Science, STACS 2004, in: Lecture Notes in Computer Science, vol. 2996, 2004, pp. 129–140] and [D. Reidenbach, On the learnability of E-pattern languages over small alphabets, in: Proc. 17th Annual Conference on Learning Theory, COLT 2004, in: Lecture Notes in Artificial Intelligence, vol. 3120, 2004, pp. 140–154].

E-mail address: [D.Reidenbach@lboro.ac.uk](mailto:D.Reidenbach@lboro.ac.uk).

URL: <http://www-staff.lboro.ac.uk/~codr2/>.

pattern language of a pattern  $\alpha$  is simply the set of all morphic images of  $\alpha$  in  $\Sigma^*$ . For instance, the pattern language of the pattern  $\alpha := x_1 x_1 a b x_2$  (with  $X := \{x_1, x_2, \dots\}$  and  $\Sigma := \{a, b\}$ ) consists of all words where the prefix can be split into two occurrences of the same string, followed by the string  $ab$  and concluded by an arbitrary suffix. Consequently, the language of  $\alpha$  contains, among others, the words  $w_1 = aaaba$ ,  $w_2 = abababab$  and  $w_3 = abbb$ , whereas the following examples are not covered by  $\alpha$ :  $v_1 = ba$ ,  $v_2 = bbbbbb$ ,  $v_3 = baabab$ . With reference to elementary insights in formal language theory, it can be easily seen that various regular and nonregular languages can be described by patterns in a compact and natural way.

Basically, two kinds of definitions of pattern languages are considered in the literature: the first – introduced by Angluin [1] in 1980 and leading to so-called *NE*-pattern languages – disallows the substitution of variables with the empty word, whereas the second – given by Shinohara [36] in 1982 and resulting in *extended*, *erasing* or simply *E*-pattern languages – allows the empty substitution; thus, in our example,  $w_3$  belongs to the *E*-, but not to the *NE*-pattern language of  $\alpha$ . Remarkably, this tiny difference in the definitions strongly influences the characteristics of the resulting classes of languages. In particular, this holds for a number of elementary decision problems such as the equivalence problem, which is a complex open question for *E*-pattern languages, but can be easily solved for arbitrary *NE*-pattern languages. Further information on basic properties of pattern languages are provided by, e. g., Jiang et al. [13,14], and corresponding surveys are given by Mateescu, Salomaa [23] and Salomaa [35].

While the investigation of patterns in strings of symbols – initiated by Thue [38] in 1906 – is a classic topic in combinatorics on words (cf. Lothaire [20]), the concept of pattern languages has originally been motivated by considerations on the algorithmic problem of computing a pattern that is common to a given set of words. More precisely, pattern languages initially have been a focus of interest of *inductive inference*, and therefore the early corresponding papers have mostly studied their *infeerability* (or, as we occasionally prefer to call it, *learnability*) in the elementary model of *identification in the limit* as introduced by Gold [8] in 1967 (an approach which frequently is referred to as *Gold-style learning*). In this model, a class of languages is said to be *infeerable from positive data* if and only if a computable device (the so-called *learning strategy*) – that reads growing initial segments of texts (an arbitrary stream of words which, in the limit, fully enumerates the language) – after finitely many steps converges for every language and for every corresponding text to a distinct output exactly representing the given language. In other words, the learning strategy is expected to extract a complete description of a (potentially infinite) language from finitely many examples for this language. According to Gold [8], this task is too challenging for many well-known classes of formal languages: All superfinite classes of languages – i.e. those classes that contain all finite and at least one infinite language – such as the regular, context-free and context-sensitive languages are not infeerable from positive data. Consequently, the number of rich classes of languages that are known to be learnable is rather small.

With regard to the infeerability of classes of pattern languages, the current state of knowledge considerably differs when comparing *NE*- with *E*-pattern languages. The full class of *NE*-pattern languages was shown to be learnable by Angluin [1,2] in her initial papers on the subject. Subsequent to this fundamental insight, there has been a variety of additional studies on the *NE* case – e.g. by Lange, Wiehagen [16], Wiehagen, Zeugmann [39], Reischuk, Zeugmann [31] and many more – discussing more specific topics such as the complexity of special learning algorithms, effects of different input data, efficient strategies for subclasses, and so on. Consequently, inductive inference of *NE*-pattern languages is a well-explored and well-understood topic (for a survey, see Shinohara, Arikawa [37]). Contrary to this, many basic questions on the learnability of classes of *E*-pattern languages are still open. In particular, most previous corresponding papers do not tackle the infeerability of the full class of *E*-pattern languages, but they merely present positive results on subclasses: Shinohara [36] proves the learnability of the class of *E*-pattern languages generated by *regular* patterns, which are characterised by the fact that every variable in  $X$  occurs at most once in such a pattern. This result is extended by Mitchell [24] to the class of so-called *quasi-regular* *E*-pattern languages, that are generated by patterns  $\alpha$  for which there exists an  $n \in \mathbb{N}$  such that every variable in  $X$  occurs either 0 or exactly  $n$  times in  $\alpha$ . Furthermore, Wright [40] indirectly shows the infeerability of the class of *E*-pattern languages generated by patterns over an arbitrary terminal alphabet  $\Sigma$  and any *finite* set  $X$  of variables. Finally, Reidenbach [30] also gives a positive learnability result on a nontrivial (yet minor) class of *E*-pattern languages. The main achievement of [30], however, is the first major insight into the infeerability of the full class of *E*-pattern languages, as it shows that the class of *terminal-free* *E*-pattern languages (generated by patterns that do not contain any terminal symbol) over a binary alphabet is *not* infeerable from positive data, which by definition implies the same outcome for the full class of *E*-pattern languages over such an alphabet. In this context note that Mitchell [24] also provides some results on the full class of *E*-pattern languages, as he proves its learnability for the cases  $|\Sigma| = 1$  and  $|\Sigma| = \infty$ ; since unary and

infinite alphabets substantially facilitate inferrability (and normally are not considered overly interesting), this result is merely of marginal significance, though.

In the present paper we continue the research on the learnability of E-pattern languages. In Section 4 we show that the class of terminal-free E-pattern languages is not inferrable from positive data if *and only if* the terminal alphabet is binary. Thus, we prove that the result by Reidenbach [30] on binary alphabets is unique, so that the learnability of terminal-free E-pattern languages is discontinuous with respect to the alphabet size. In Section 5 we demonstrate that the inferrability of the full class of E-pattern languages differs from that of terminal-free E-pattern languages: for alphabets of size 3 and 4, the said class is not inferrable from positive data. To this end, we introduce a non-learnable subclass of general E-pattern languages, generated by the so-called *quasi-terminal-free* patterns. We are unable to provide an answer for alphabets with five or more letters.

Our proofs do not directly discuss the (non-)existence of suitable learning strategies, but they are based on a structural argument on learnable classes of languages given by Angluin [2]. Using this as a tool, we show in Section 3 that, with regard to the class of terminal-free E-pattern languages, the problem of its inferrability is equivalent to a question on the ambiguity of morphisms in word monoids. Additionally, we provide and utilise a characterisation of the shortest generators of terminal-free E-pattern languages which, as a side-effect, shows that the set of these patterns equals the set of strings that are not a fixed point of a nontrivial morphism. Consequently, our reasoning on terminal-free E-pattern languages does not only yield insights into their learnability, but it also leads to some statements of intrinsic interest on combinatorics on words. Contrary to this, besides their learning theoretical implication, our considerations on general E-pattern languages mainly show that the ambiguity of terminal-preserving morphisms strongly differs from that of common morphisms.

## 2. Definitions and preliminary results

In order to keep this paper largely self-contained we now introduce a number of definitions and basic properties. For standard mathematical notions and recursion-theoretic terms not defined explicitly, we refer the reader to Rogers [32]; for unexplained aspects of formal language theory, Rozenberg, Salomaa [34] may be consulted.

$\mathbb{N}_0$  is the set of natural numbers  $\{0, 1, 2, \dots\}$  and  $\mathbb{N} := \mathbb{N}_0 \setminus \{0\}$ . The symbol  $\subseteq$  denotes the subset relation,  $\subset$  the proper subset relation,  $\supseteq$  the superset relation,  $\supset$  the proper superset relation, and  $\emptyset$  the empty set. An *alphabet*  $\mathcal{A}$  is an enumerable set of symbols. A *string* (over  $\mathcal{A}$ ) is a finite sequence of symbols which are contained in  $\mathcal{A}$ , i.e.  $s = a_1 a_2 \dots a_n$  with  $n \in \mathbb{N}_0$  and  $a_i \in \mathcal{A}$ ,  $1 \leq i \leq n$ . The size of a set  $A$  is denoted by  $|A|$  and the length of a string  $s$  by  $|s|$ . For the string that results from the  $n$ -fold concatenation of a string  $s$  we write  $s^n$ . The string of length 0 is called the *empty* string; it is denoted by  $\varepsilon$ . For an arbitrary alphabet  $\mathcal{A}$ ,  $\mathcal{A}^+$  denotes the set of all non-empty strings over  $\mathcal{A}$ , and  $\mathcal{A}^* := \mathcal{A}^+ \cup \{\varepsilon\}$ . Any set  $L \subseteq \mathcal{A}^*$  is a *language* (over  $\mathcal{A}$ ).

We call a string  $t \in \mathcal{A}^*$  a *substring* of a string  $s \in \mathcal{A}^*$  if, for some  $r_1, r_2 \in \mathcal{A}^*$ ,  $s = r_1 t r_2$ . In addition to this, if  $t$  is a substring of  $s$  then we say that  $s$  *contains*  $t$  and, conversely, that  $t$  *occurs* in  $s$ . The notation  $|s|_t$  stands for the number of occurrences of the substring  $t$  in the string  $s$ . If, for some  $s, t_1, t_2 \in \mathcal{A}^*$ ,  $s = t_1 t_2$  then  $t_1$  is a *prefix* of  $s$  and  $t_2$  is a *suffix* of  $s$ . Additionally, we use the notations  $s = \dots t \dots$  if  $t$  is a substring of  $s$ ,  $s = t \dots$  if  $t$  is a prefix of  $s$ , and  $\dots = s$  if  $t$  is a suffix of  $s$ . In contrast to this, if we wish to omit in our presentation some parts of a canonically given string then we henceforth use the symbol  $[\dots]$ , i.e., e.g.,  $s = a_1 a_2 [\dots] a_5$  stands for  $s = a_1 a_2 a_3 a_4 a_5$ . For any string  $s$  that contains at least one occurrence of a symbol  $a$  we define the following substrings:  $[s/a]$  is the prefix of  $s$  up to (but not including) the leftmost occurrence of the letter  $a$  and  $[a \setminus s]$  is the suffix of  $s$  beginning with the first letter that is to the right of the leftmost occurrence of  $a$  in  $s$ . Thus, the specified substrings satisfy  $s = [s/a] a [a \setminus s]$ ; e.g., for  $s := bcaab$ , the substring  $[s/a]$  equals  $bc$  and  $[a \setminus s]$  equals  $ab$ .

For any two alphabets  $\mathcal{A}, \mathcal{B}$ , a *morphism*  $f : \mathcal{A}^* \rightarrow \mathcal{B}^*$  is a mapping that is compatible with the concatenation, i.e., for each pair of strings  $s_1, s_2 \in \mathcal{A}^*$ , the morphism  $f$  satisfies  $f(s_1 s_2) = f(s_1) f(s_2)$ . Hence, a morphism is fully defined as soon as it is declared for all symbols in  $\mathcal{A}$ . Note that, for every morphism  $f$ ,  $f(\varepsilon) = \varepsilon$ . For the *composition* of two morphisms  $f, g$  we write  $g \circ f$ , i.e., for every  $s \in \mathcal{A}^*$ ,  $g \circ f(s) = g(f(s))$ .

We proceed with the pattern specific terminology.  $\Sigma$  is a finite or infinite alphabet of *terminal* symbols (or, alternatively: *letters*). Henceforth, we restrict the use of the term “alphabet” to terminal alphabets, and we use lower case letters in typewriter font, e.g.  $a, b, c$ , as terminal symbols. For unspecified terminal symbols we use upper case letters in typewriter font, such as  $A$ . A string over  $\Sigma$  is called a (*terminal*) *word*. Words normally are named as  $u, v$ , or  $w$ . Additionally, we introduce the set  $X := \{x_i \mid i \in \mathbb{N}\}$ ,  $\Sigma \cap X = \emptyset$ , and we call any symbol  $x_i$  a *variable*. Thus,

for every  $k, k' \in \mathbb{N}$ ,  $x_k = x_{k'}$  if and only if  $k = k'$ . Contrary to this, for every  $j \geq 1$ , we use the symbol  $y_j \in X$  as an *unspecified variable*, i.e. there may exist indices  $k, k'$  such that  $k \neq k'$ , but  $y_k = y_{k'}$ .

A *pattern (over  $\Sigma$ )* is a non-empty string over  $\Sigma \cup X$ , a *terminal-free pattern* is a non-empty string over  $X$ ; for naming patterns we use lower case letters from the beginning of the Greek alphabet such as  $\alpha, \beta, \gamma$ .  $\text{var}(\alpha)$  denotes the set of all variables occurring in a pattern  $\alpha$  and  $\text{term}(\alpha)$  the set of all terminals in  $\alpha$ . The pattern  $\text{tf}(\alpha)$  is derived from the pattern  $\alpha$  by removing all terminal symbols from  $\alpha$ ; e.g.,  $\text{tf}(x_1x_1ax_2b) = x_1x_1x_2$ . We write  $\text{Pat}_\Sigma$  for the set of all patterns, and we use  $\text{Pat}$  instead of  $\text{Pat}_\Sigma$  if  $\Sigma$  is understood; moreover,  $\text{Pat}_{\text{tf}}$  denotes the set of all terminal-free patterns.

We call patterns  $\alpha, \beta \in \text{Pat}$  *similar* if and only if  $\alpha = \alpha_0u_1\alpha_1u_2[\dots]\alpha_{m-1}u_m\alpha_m$  and  $\beta = \beta_0u_1\beta_1u_2[\dots]\beta_{m-1}u_m\beta_m$  with  $m \in \mathbb{N}_0$ ,  $\alpha_i, \beta_i \in X^+$  for  $1 \leq i < m$ ,  $\alpha_0, \beta_0, \alpha_m, \beta_m \in X^*$  and  $u_i \in \Sigma^+$  for  $i \leq m$ ; in other words, we call patterns similar if and only if their terminal substrings coincide.

For any alphabet  $\Sigma$ , a morphism  $\phi : (\Sigma \cup X)^* \longrightarrow (\Sigma \cup X)^*$  is called *terminal-preserving* provided that, for every  $A \in \Sigma$ ,  $\phi(A) = A$ . A terminal-preserving morphism  $\sigma : (\Sigma \cup X)^* \longrightarrow \Sigma^*$  is a *substitution*, and an *inverse substitution*  $\bar{\sigma}$  is a morphism satisfying  $\bar{\sigma} : \Sigma^* \longrightarrow X^*$ . The *E-pattern language*  $L_\Sigma(\alpha)$  of a pattern  $\alpha$  is defined as the set of all  $w \in \Sigma^*$  such that  $\sigma(\alpha) = w$  for some substitution  $\sigma$ . For any word  $w = \sigma(\alpha)$  we say that  $\sigma$  *generates*  $w$ , and for any language  $L = L_\Sigma(\alpha)$  we say that  $\alpha$  *generates*  $L$ . If there is no need to give emphasis to the concrete shape of  $\Sigma$  we denote the E-pattern language of a pattern  $\alpha$  simply as  $L(\alpha)$ . We use  $\text{ePAT}_\Sigma$  (or  $\text{ePAT}$  for short) as an abbreviation for the full class of E-pattern languages and  $\text{ePAT}_{\text{tf}, \Sigma}$  (or  $\text{ePAT}_{\text{tf}}$  for short) for the class of all terminal-free E-pattern languages over  $\Sigma$ .

We designate a pattern  $\alpha$  as *succinct (with respect to an alphabet  $\Sigma$ )* if and only if  $|\alpha| \leq |\beta|$  for all patterns  $\beta$  with  $L_\Sigma(\beta) = L_\Sigma(\alpha)$ , and we call  $\alpha$  *prolix (with respect to an alphabet  $\Sigma$ )* if and only if it is not succinct. The pattern  $\beta := x_1x_2x_1x_2$ , for instance, generates the same language as the pattern  $\alpha := x_1x_1$ , and therefore  $\beta$  is prolix;  $\alpha$  is succinct because there does not exist any shorter pattern than  $\alpha$  that exactly describes its language.

We denote a word  $w$  as *ambiguous (with respect to a pattern  $\alpha$ )* if and only if there exist two substitutions  $\sigma$  and  $\tau$  such that  $\sigma(\alpha) = w = \tau(\alpha)$ , but  $\sigma(x_i) \neq \tau(x_i)$  for some  $x_i \in \text{var}(\alpha)$ . We call  $w$  *unambiguous (with respect to  $\alpha$ )* if and only if there is exactly one substitution  $\sigma$  with  $\sigma(\alpha) = w$ . The word  $w_1 := \text{aaba}$ , for instance, is ambiguous with respect to the pattern  $\alpha := x_1ax_2$  since it can be generated by several substitutions, such as  $\sigma$  and  $\tau$  with  $\sigma(x_1) := \text{a}$ ,  $\sigma(x_2) := \text{ba}$  and  $\tau(x_1) := \varepsilon$ ,  $\tau(x_2) := \text{aba}$ . Contrary to this, the word  $w_2 := \text{abb}$  is unambiguous with respect to  $\alpha$ .

We now proceed with the description of some crucial decision problems on E-pattern languages: Let  $\text{ePAT}_\star$  be any set of E-pattern languages. We say that the *inclusion problem* for  $\text{ePAT}_\star$  is *decidable* if and only if there exists a computable function which, given two arbitrary patterns  $\alpha, \beta$  with  $L(\alpha), L(\beta) \in \text{ePAT}_\star$ , decides on whether or not  $L(\alpha) \subseteq L(\beta)$ . Correspondingly, the *equivalence problem* for  $\text{ePAT}_\star$  is decidable if and only if there exists another computable function which for every pair of patterns  $\alpha, \beta$  with  $L(\alpha), L(\beta) \in \text{ePAT}_\star$  decides on whether or not  $L(\alpha) = L(\beta)$ . Obviously, the decidability of the inclusion implies the decidability of the equivalence. The inclusion problem is known to be undecidable provided that the class of E-pattern languages over all alphabets is considered (cf. Jiang et al. [14]). Within the scope of the present paper, however, we solely consider classes of E-pattern languages over some *fixed* alphabet. With regard to this problem, for every finite  $\Sigma$  with  $|\Sigma| \geq 2$ , the decidability of the inclusion problem for  $\text{ePAT}_\Sigma$  is open, and so is the decidability of the equivalence problem for  $\text{ePAT}_\Sigma$ . Still, we can rely on the following *sufficient* criterion for the inclusion of E-pattern languages:

**Theorem 1** (Jiang et al. [13]). *Let  $\Sigma$  be an alphabet, and let  $\alpha, \beta \in \text{Pat}_\Sigma$ . Then  $L_\Sigma(\beta) \subseteq L_\Sigma(\alpha)$  if there exists a terminal-preserving morphism  $\phi : (\Sigma \cup X)^* \longrightarrow (\Sigma \cup X)^*$  with  $\phi(\alpha) = \beta$ .*

Note that it can be effectively tested whether there exists a morphism mapping a string onto another (see our remarks on the membership problem given below).

If the patterns under consideration are terminal-free then the above criterion additionally is *necessary* for the inclusion of their languages, and therefore the inclusion and the equivalence problem are decidable for  $\text{ePAT}_{\text{tf}}$ :

**Theorem 2** (Filè [6], Jiang et al. [14]). *Let  $\Sigma$  be an alphabet,  $|\Sigma| \geq 2$ , and let  $\alpha, \beta \in \text{Pat}_{\text{tf}}$ . Then  $L_\Sigma(\beta) \subseteq L_\Sigma(\alpha)$  if and only if there exists a morphism  $\phi : X^* \longrightarrow X^*$  with  $\phi(\alpha) = \beta$ .*

In Section 5.1, we present an extension of Theorem 2 that is due to Ohlebusch, Ukkonen [25].

This paper exclusively deals with language theoretical properties of E-pattern languages. Both motivation and interpretation of our examination, however, are based on learning theory, and therefore we consider it useful to

provide an adequate background. To this end, we now introduce our notions on Gold's learning model (cf. Gold [8]) of *identification in the limit* and begin with a specification of the objects to be learned. In this regard, we restrict ourselves to any *indexable class of non-empty languages*; a class  $\mathcal{L}$  of languages is indexable if and only if there exists an *indexed family (of non-empty recursive languages)*  $(L_i)_{i \in \mathbb{N}_0}$  such that  $\mathcal{L} = \{L_i \mid i \in \mathbb{N}_0\}$ —this means that the *membership* is uniformly decidable for  $(L_i)_{i \in \mathbb{N}_0}$ , i.e. there is a total and computable function which, given any pair of an index  $i \in \mathbb{N}_0$  and a word  $w \in \Sigma^*$ , decides on whether or not  $w \in L_i$ . Concerning the learner's input, we exclusively consider inference from positive data given as *text*. A text for an arbitrary language  $L$  is any total function  $t : \mathbb{N}_0 \rightarrow \Sigma^*$  satisfying  $\{t(n) \mid n \in \mathbb{N}_0\} = L$ . For any text  $t$ , any  $n \in \mathbb{N}_0$  and a symbol  $\diamond \notin \Sigma$ ,  $t^n \in (\Sigma \cup \{\diamond\})^+$  is a coding of the first  $n + 1$  values of  $t$ , i.e.  $t^n := t(0) \diamond t(1) \diamond t(2) [\dots] \diamond t(n)$ . Last, the learner and the learning goal need to be explained: Let the *learner* (or: the *learning strategy*)  $S$  be a computable function that, for any given text  $t$ , successively reads  $t^0, t^1, t^2$ , etc. and returns a corresponding stream of natural numbers  $S(t^0), S(t^1), S(t^2)$ , and so on. For a language  $L_j$  and a text  $t$  for  $L_j$ , we say that  $S$  *identifies*  $L_j$  *from*  $t$  if and only if there exist natural numbers  $n_0$  and  $j'$  such that, for every  $n \geq n_0$ ,  $S(t^n) = j'$  and, additionally,  $L_{j'} = L_j$ . An indexed family  $(L_i)_{i \in \mathbb{N}_0}$  is *learnable (in the limit)* (or: *inferred from positive data*) if and only if there is a learning strategy  $S$  identifying each language in  $(L_i)_{i \in \mathbb{N}_0}$  from any corresponding text. Finally, we call an indexable class  $\mathcal{L}$  of languages learnable (in the limit) or inferred from positive data if and only if there is a learnable indexed family  $(L_i)_{i \in \mathbb{N}_0}$  with  $\mathcal{L} = \{L_i \mid i \in \mathbb{N}_0\}$ .

For the sake of convenience, the specific learning model given above – that is largely based on Angluin [2] – is just a special case of Gold's learning model. For insights into numerous variations of Gold's model, see e.g. Zeugmann, Lange [41] and Lange [15]. In this context, we wish to note that our results in the subsequent sections hold in several other learning models as well, so that they are not as limited as suggested by our choice of the learning model. This fact, that is a consequence of our proof technique (based on Theorem 3) and various insights into the relations between prominent learning models, can be verified referring to, e.g., Jain et al. [12], Baliga et al. [3] and Lange, Zilles [17, 18].

Angluin [2] introduces some criteria that reduce the learnability of indexed families to a particular language theoretical aspect and thereby facilitate our approach to learnability questions. Our subsequent reasoning shall be based on the most fundamental one, which characterises those indexed families that are inferred from positive data (combining Condition 1 and Theorem 1 of the referenced paper):

**Theorem 3** (Angluin [2]). *Let  $(L_i)_{i \in \mathbb{N}_0}$  be an indexed family of non-empty recursive languages. Then  $(L_i)_{i \in \mathbb{N}_0}$  is inferred from positive data if and only if there exists an effective procedure which, for every  $j \in \mathbb{N}_0$ , enumerates a set  $T_j$  such that*

- $T_j$  is finite,
- $T_j \subseteq L_j$ , and
- there does not exist a  $j' \in \mathbb{N}_0$  with  $T_j \subseteq L_{j'} \subset L_j$ .

If there exists a set  $T_j$  satisfying the conditions of Theorem 3 then it is called a *telltale (for  $L_j$ ) (with respect to  $(L_i)_{i \in \mathbb{N}_0}$ )*.

The importance of telltales – that, at first glance, do not show any connection to the learning model – is caused by the need of avoiding *overgeneralisation* in the inference process, i.e. the case that the strategy outputs an index of a language which is a proper superset of the language to be learned and therefore, as the input consists of positive data only, is unable to detect its mistake. Thus, for every language  $L_j$  in a learnable indexed family it is characteristic that it contains a finite set of words which, in the context of the indexed family, may be interpreted as a signal distinguishing the language from all languages that are subsets of  $L_j$ .

With regard to classes of E-pattern languages, Theorem 3 is applicable because ePAT is an indexable class of non-empty languages. This is evident as, first, every E-pattern language is non-empty, second, a recursive enumeration of all patterns can be constructed with little effort and, third, the decidability of the membership problem for any pattern  $\alpha \in \text{Pat}$  and word  $w \in \Sigma^*$  is guaranteed since the search space for a successful substitution of  $\alpha$  is bounded by the length of  $w$  and the number of different letters occurring in  $w$ . Note that the membership problem for pattern languages is NP-complete (cf. Ehrenfeucht, Rozenberg [5], Angluin [1], Jiang et al. [13]).

Thus, we can conclude this section with a naming for a particular type of patterns that is introduced in [30] and that directly aims at the content of Theorem 3: A pattern  $\beta$  is a *passe-partout (for a pattern  $\alpha$  and a finite set  $W$  of*



words) if and only if  $W \subseteq L(\beta)$  and  $L(\beta) \subset L(\alpha)$ . Consequently, if there exists such a passe-partout  $\beta$  then  $W$  is not a telltale for  $L(\alpha)$  with respect to any class of E-pattern languages containing both  $L(\alpha)$  and  $L(\beta)$ .

### 3. Preparatory technical considerations of intrinsic interest

Before we examine the learnability of classes of E-pattern languages in Gold's model of identification in the limit we give two characteristic criteria on the subject: in [Theorem 7](#) we determine the structural properties of succinct terminal-free patterns, and [Theorem 10](#) describes the shape of telltales for terminal-free E-pattern languages with respect to  $\text{ePAT}_{\text{tf}}$ . Both of these theorems are vital for our examination of the learnability of  $\text{ePAT}_{\text{tf}, \Sigma}$  in case of  $|\Sigma| \geq 3$  (see [Section 4](#)). Furthermore, these theorems – the methodology of which can be subsumed under the field of combinatorics on words (cf., e.g., Lothaire [20], Choffrut, Karhumäki [4]) – show some fundamental and rather unexpected analogies between E-pattern languages and other classic topics in discrete mathematics.

We begin this section with three lemmata describing basic combinatorial observations on morphisms in free monoids, that are needed at several stages of our paper.

**Lemma 4.** *Let  $\alpha, \beta \in \text{Pat}_{\text{tf}}$ . Let  $\phi, \psi : X^* \rightarrow X^*$  be morphisms with  $\phi(\alpha) = \beta$  and  $\psi(\beta) = \alpha$ . Then either, for every  $x_j \in \text{var}(\alpha)$ ,  $\psi(\phi(x_j)) = x_j$  or there exists an  $x_{j'} \in \text{var}(\alpha)$  such that  $|\psi(\phi(x_{j'}))| \geq 2$  and  $x_{j'} \in \text{var}(\psi(\phi(x_{j'})))$ .*

We call any  $x_{j'}$  satisfying these two conditions an *anchor variable* (with respect to  $\phi$  and  $\psi$ ).

**Proof.** Let  $\alpha := y_1 y_2 y_3 [\dots] y_m$ ; then  $\beta = \phi(y_1) \phi(y_2) \phi(y_3) [\dots] \phi(y_m)$ . Let  $y_{k_0}$  be the leftmost variable such that  $\psi(\phi(y_{k_0})) \neq y_{k_0}$ . Now assume to the contrary that there is no anchor variable in  $\alpha$ . Then  $\psi(\phi(y_{k_0}))$  necessarily equals  $\varepsilon$  as otherwise  $\psi(\beta) \neq \alpha$ . Hence,  $|\psi(\phi(y_1)) \psi(\phi(y_2)) \psi(\phi(y_3)) [\dots] \psi(\phi(y_{k_0}))| = k_0 - 1$ , and, as there is no anchor variable in  $\alpha$ ,  $|\psi(\phi(y_1)) \psi(\phi(y_2)) \psi(\phi(y_3)) [\dots] \psi(\phi(y_k))| \leq k - 1$  for every  $k > k_0$ . Consequently,  $|\psi(\beta)| < |\alpha|$  and therefore  $\psi(\beta) \neq \alpha$ . This contradiction proves the lemma.  $\square$

From [Lemma 4](#) we can immediately conclude the following fact:

**Lemma 5.** *Let  $\alpha, \beta \in \text{Pat}_{\text{tf}}$ . Let  $\phi, \psi : X^* \rightarrow X^*$  be morphisms with  $\phi(\alpha) = \beta$  and  $\psi(\beta) = \alpha$ . Then either, for every  $x_j \in \text{var}(\alpha)$ ,  $\psi(\phi(x_j)) = x_j$  or there exists an  $x_{j'} \in \text{var}(\alpha)$  such that  $\psi(\phi(x_{j'})) = \varepsilon$ .*

The last of our initial lemmata discusses a property of those morphisms which – according to [Theorem 2](#) when using it as a criterion on the equivalence of terminal-free E-pattern languages – map a succinct pattern  $\alpha$  and a prolix pattern  $\beta$  generating the same language onto each other:

**Lemma 6.** *Let  $\alpha, \beta \in \text{Pat}_{\text{tf}}$ ,  $\alpha$  succinct. Let  $\phi, \psi : X^* \rightarrow X^*$  be morphisms with  $\phi(\alpha) = \beta$  and  $\psi(\beta) = \alpha$ . Then, for every  $x_j \in \text{var}(\alpha)$ ,  $\psi(\phi(x_j)) = x_j$ .*

**Proof.** Assume to the contrary that there exists an  $x_j \in \text{var}(\alpha)$  with  $\psi(\phi(x_j)) \neq x_j$ . Then, according to [Lemma 5](#), there is an  $x_{j'} \in \text{var}(\alpha)$  such that  $\psi(\phi(x_{j'})) = \varepsilon$ . We now regard the morphism

$$\phi'(x_j) := \begin{cases} \varepsilon, & j = j', \\ x_j, & \text{else,} \end{cases}$$

$x_j \in \text{var}(\alpha)$ , and define  $\alpha' := \phi'(\alpha)$ . Hence,  $|\alpha'| < |\alpha|$  and, due to [Theorem 1](#),  $L(\alpha') \subseteq L(\alpha)$ . Moreover – since, for every  $x_k \in \text{var}(\phi(x_{j'}))$ ,  $\psi(x_k) = \varepsilon$  – we can state  $\psi(\phi(\alpha')) = \alpha$  and therefore  $L(\alpha') \supseteq L(\alpha)$ . Since  $|\alpha'| < |\alpha|$ , the resulting equality of  $L(\alpha)$  and  $L(\alpha')$  contradicts the condition of  $\alpha$  being succinct.  $\square$

We now can give the first main result of the present section. It shows that a pattern  $\alpha$  is succinct if and only if there is no particular decomposition of  $\alpha$ :

**Theorem 7.** *Let  $\Sigma$  be an alphabet,  $|\Sigma| \geq 2$ . Then a pattern  $\alpha \in \text{Pat}_{\text{tf}}$  is prolix with respect to  $\Sigma$  if and only if there exists a decomposition*

$$\alpha = \beta_0 \gamma_1 \beta_1 \gamma_2 \beta_2 [\dots] \beta_{n-1} \gamma_n \beta_n$$

for an  $n \geq 1$ , arbitrary  $\beta_k \in X^*$  and  $\gamma_k \in X^+$ ,  $k \leq n$ , such that

- (i) for every  $k$ ,  $1 \leq k \leq n$ ,  $|\gamma_k| \geq 2$ ,  
(ii) for every  $k$ ,  $1 \leq k \leq n$ , and for every  $k'$ ,  $0 \leq k' \leq n$ ,  $\text{var}(\gamma_k) \cap \text{var}(\beta_{k'}) = \emptyset$ , and  
(iii) for every  $k$ ,  $1 \leq k \leq n$ , there exists a  $y_k \in \text{var}(\gamma_k)$  such that  $|\gamma_k|_{y_k} = 1$  and, for every  $k'$ ,  $1 \leq k' \leq n$ , if  $y_k \in \text{var}(\gamma_{k'})$  then  $\gamma_k = \gamma_{k'}$ .

**Proof.** We first prove the *if* part of the theorem. Hence, let  $\alpha \in \text{Pat}_{\text{tf}}$  be a pattern such that there exists a decomposition satisfying conditions (i), (ii), and (iii). We show that then there exist a pattern  $\delta \in \text{Pat}_{\text{tf}}$  and two morphisms  $\phi$  and  $\psi$  with  $|\delta| < |\alpha|$ ,  $\phi(\delta) = \alpha$ , and  $\psi(\alpha) = \delta$ . Thus, we use [Theorem 2](#) as a criterion for the equivalence of E-pattern languages.

We define

$$\delta := \beta_0 y_1 \beta_1 y_2 \beta_2 [\dots] \beta_{n-1} y_n \beta_n,$$

where  $y_k$  is derived from condition (iii) for every  $k \leq n$ . Then  $|\delta| < |\alpha|$  because of condition (i). As a first morphism we define

$$\phi(x_j) := \begin{cases} \gamma_k, & x_j = y_k \text{ for a } k, 1 \leq k \leq n, \\ x_j, & \text{else,} \end{cases}$$

$x_j \in \text{var}(\delta)$ . Because of conditions (ii) and (iii),  $\phi$  really is a morphism; obviously,  $\phi(\delta) = \alpha$ . The second morphism reads

$$\psi(x_j) := \begin{cases} \varepsilon, & x_j \in \text{var}(\gamma_k) \text{ for a } k, 1 \leq k \leq n, \text{ and } x_j \neq y_k, \\ x_j, & \text{else,} \end{cases}$$

$x_j \in \text{var}(\alpha)$ . Consequently,  $\psi(\alpha) = \delta$  and therefore  $L(\alpha) = L(\delta)$ . Since  $\delta$  is shorter than  $\alpha$ ,  $\alpha$  is prolix.

For the *only if* part assume that  $\alpha \in \text{Pat}_{\text{tf}}$  is prolix. We show that this assumption implies the existence of a decomposition of  $\alpha$  satisfying conditions (i), (ii), and (iii): If  $\alpha$  is prolix then there exist morphisms  $\phi, \psi : X^* \rightarrow X^*$  and a succinct pattern  $\delta \in \text{Pat}_{\text{tf}}$  such that  $|\delta| < |\alpha|$ ,  $\phi(\delta) = \alpha$ , and  $\psi(\alpha) = \delta$ . This leads to

**Claim 1.** For every  $x_j \in \text{var}(\delta)$ ,  $\phi(x_j) \neq \varepsilon$ .

**Proof (Claim 1):** Since  $\delta, \alpha, \phi$  and  $\psi$  satisfy the conditions of [Lemma 6](#) we may conclude that, for every  $x_j \in \text{var}(\delta)$ ,  $\psi(\phi(x_j)) = x_j \neq \varepsilon$ . Consequently,  $\phi(x_j) \neq \varepsilon$ .  $\square$  (Claim 1)

In addition, [Lemma 6](#) also is the decisive tool for the proof of

**Claim 2.** For every  $x_j \in \text{var}(\delta)$  there is an  $x_{j'} \in \text{var}(\alpha)$  such that  $x_{j'} \in \text{var}(\phi(x_j))$  and  $|\delta|_{x_j} = |\alpha|_{x_{j'}}$ .

**Proof (Claim 2):** Assume to the contrary that there is an  $x_j \in \text{var}(\delta)$  such that, for every  $x_i \in \text{var}(\alpha)$ ,  $x_i \notin \text{var}(\phi(x_j))$  or  $|\alpha|_{x_i} \neq |\delta|_{x_j}$ . Consequently, for every  $x_i \in \text{var}(\phi(x_j))$ ,  $|\alpha|_{x_i} > |\delta|_{x_j}$  since necessarily  $|\alpha|_{x_i} \geq |\delta|_{x_j}$ . Therefore, for every  $x_i \in \text{var}(\phi(x_j))$ ,  $x_j \notin \text{var}(\psi(x_i))$ , and, thus,  $\psi(\phi(x_j)) \neq x_j$ . This contradicts [Lemma 6](#).  $\square$  (Claim 2)

We now regard the following subsets of  $\text{var}(\delta)$ :  $X_1 := \{x_j \in \text{var}(\delta) \mid |\phi(x_j)| = 1\}$  and  $X_2 := \text{var}(\delta) \setminus X_1$ . This partition of  $\text{var}(\delta)$  leads to a particular decomposition of  $\delta$ :

$$\begin{aligned} \delta &= \bar{\beta}_0 \underbrace{\bar{y}_1 \bar{y}_2 [\dots] \bar{y}_{s_1}}_{\bar{\gamma}_1} \\ &\quad \bar{\beta}_1 \underbrace{\bar{y}_{s_1+1} \bar{y}_{s_1+2} [\dots] \bar{y}_{s_2}}_{\bar{\gamma}_2} \bar{\beta}_2 \\ &\quad [\dots] \\ &\quad \bar{\beta}_{m-1} \underbrace{\bar{y}_{s_{m-1}+1} \bar{y}_{s_{m-1}+2} [\dots] \bar{y}_{s_m}}_{\bar{\gamma}_m} \bar{\beta}_m \end{aligned}$$

with

- $m \in \mathbb{N}$ ,
- for every  $i$ ,  $1 \leq i \leq m$ ,  $|\bar{\gamma}_i| =: p_i \geq 1$ ,
- for every  $k$ ,  $1 \leq k \leq m$ ,  $s_k := \sum_{i=1}^k p_i$ ,

- for every  $j$ ,  $1 \leq j \leq s_m$ ,  $\bar{y}_j \in X_2$ , and
- $\bar{\beta}_0, \bar{\beta}_m \in X_1^*$ ,  $\bar{\beta}_1, \bar{\beta}_2, \dots, \bar{\beta}_{m-1} \in X_1^+$ .

This decomposition of  $\delta$  is unique, and, with  $\phi$ , it induces an appropriate decomposition of  $\alpha = \phi(\delta)$ :

$$\begin{aligned} \alpha = & \underbrace{\phi(\bar{\beta}_0)}_{\beta_0} \underbrace{\phi(\bar{y}_1)}_{\gamma_1} \underbrace{\varepsilon}_{\beta_1} \underbrace{\phi(\bar{y}_2)}_{\gamma_2} \underbrace{\varepsilon}_{\beta_2} [\dots] \underbrace{\varepsilon}_{\beta_{s_1-1}} \underbrace{\phi(\bar{y}_{s_1})}_{\gamma_{s_1}} \\ & \underbrace{\phi(\bar{\beta}_1)}_{\beta_{s_1}} \underbrace{\phi(\bar{y}_{s_1+1})}_{\gamma_{s_1+1}} \underbrace{\varepsilon}_{\beta_{s_1+1}} \underbrace{\phi(\bar{y}_{s_1+2})}_{\gamma_{s_1+2}} \underbrace{\varepsilon}_{\beta_{s_1+2}} [\dots] \underbrace{\varepsilon}_{\beta_{s_2-1}} \underbrace{\phi(\bar{y}_{s_2})}_{\gamma_{s_2}} \underbrace{\phi(\bar{\beta}_2)}_{\beta_{s_2}} \\ & [\dots] \\ & \underbrace{\phi(\bar{\beta}_{m-1})}_{\beta_{s_{m-1}}} \underbrace{\phi(\bar{y}_{s_{m-1}+1})}_{\gamma_{s_{m-1}+1}} \underbrace{\varepsilon}_{\beta_{s_{m-1}+1}} \underbrace{\phi(\bar{y}_{s_{m-1}+2})}_{\gamma_{s_{m-1}+2}} \underbrace{\varepsilon}_{\beta_{s_{m-1}+2}} [\dots] \underbrace{\varepsilon}_{\beta_{s_m-1}} \underbrace{\phi(\bar{y}_{s_m})}_{\gamma_{s_m}} \underbrace{\phi(\bar{\beta}_m)}_{\beta_{s_m}}. \end{aligned}$$

Then, for this decomposition of  $\alpha$ , Claim 1 and the definition of  $X_2 \subseteq \text{var}(\delta)$  prove condition (i). Condition (ii) follows from Claim 2 and the statement that, for every  $k \leq m$ ,  $\bar{\beta}_k \in X_1^*$ . Finally, condition (iii) is satisfied because of Claim 2 and the fact that the above decomposition is given by a morphism, leading to  $\gamma_k = \phi(\bar{y}_k) = \phi(\bar{y}_{k'}) = \gamma_{k'}$  for every  $k, k'$  with  $\bar{y}_k = \bar{y}_{k'}$ .  $\square$

Note that Theorem 7 does not imply a new decidability result on the equivalence of E-pattern languages. In fact, the decidability of the equivalence problem follows from the result by Jiang et al. [14] cited in Theorem 2. Nevertheless, the above theorem might allow for a more efficient decision procedure than those known so far, and it is crucial for the proof of Lemma 24.

The following example illustrates the decomposition introduced in Theorem 7:

**Example 8.** Note that there may be different decompositions of one and the same prolix pattern. For the subsequent prolix patterns, we nevertheless give only one of them.

- A pattern  $\alpha$  is prolix if it contains a variable  $x_j$  with  $|\alpha|_{x_j} = 1$  since, in that case, we can always – possibly among other options – rely on the decomposition satisfying  $\beta_0 = \beta_1 = \varepsilon$  and  $\gamma_1 = \alpha$ .
- The pattern  $x_1x_2x_1x_2$  is prolix with  $\gamma_1 = \gamma_2 = x_1x_2$  and  $\beta_0 = \beta_1 = \beta_2 = \varepsilon$ .
- The patterns  $x_1x_2x_2x_1x_2x_2x_2$  and  $x_1x_2x_1x_3x_4x_2x_4x_3$  are succinct because no variable for each of its occurrences has the same “environment” (i.e. a suitable  $\gamma$ ) of length greater or equal 2 so that this environment does not share any of its variables with any potential  $\beta$ ; thus, there is no decomposition of these patterns satisfying the conditions of Theorem 7.
- The pattern  $x_1x_2x_1x_2x_3x_3x_2x_4x_4x_5x_3x_2x_4x_4x_5$  is prolix with  $\gamma_1 = \gamma_2 = x_1x_2$ ,  $\gamma_3 = \gamma_4 = x_2x_4x_4x_5$ ,  $\beta_0 = \beta_1 = \beta_4 = \varepsilon$ ,  $\beta_2 = x_3x_3$ ,  $\beta_3 = x_3$ .

Additional succinct terminal-free example patterns are presented in Examples 11–15 and in Section 4.2.

Although it has no immediate impact on the learnability of pattern languages, we consider it noteworthy that Theorem 7 reveals a fundamental analogy between terminal-free E-pattern languages and *finite fixed points of nontrivial morphisms*, i.e. those strings  $s$  for which there exists a morphism  $\phi$  with  $\phi(s) = s$  and  $\phi(a) \neq a$  for some symbol  $a$  in  $s$ . As shown by Head [11], the set of finite fixed points is characterised by the existence of the same decomposition as the one identified by Theorem 7 and therefore a pattern is prolix if and only if it is a fixed point:

**Corollary 9.** Let  $\Sigma$  be an alphabet,  $|\Sigma| \geq 2$ , and let  $\alpha \in \text{Pat}_{\text{tf}}$ . Then  $\alpha$  is prolix with respect to  $\Sigma$  if and only if there exists a morphism  $\phi$  such that  $\phi(\alpha) = \alpha$  and, for some  $x_i \in \text{var}(\alpha)$ ,  $\phi(x_i) \neq x_i$ .

For additional information on fixed points of morphisms, e. g. Hamm, Shallit [9] can be consulted.

We proceed with the second main result of the present section. The following theorem gives a criterion which allows us to (effectively) decide on whether or not a given set of words is a telltale (with respect to  $\text{ePAT}_{\text{tf}}$ ) for the language of a given succinct terminal-free pattern  $\alpha$ :



**Theorem 10.** Let  $\Sigma$  be an alphabet,  $|\Sigma| \geq 2$ , and let  $\alpha \in \text{Pat}_{\text{tf}}$  be a succinct pattern. Let  $T_\alpha = \{w_1, w_2, \dots, w_n\} \subseteq L_\Sigma(\alpha)$ ,  $n \geq 1$ . Then  $T_\alpha$  is a telltale for  $L_\Sigma(\alpha)$  with respect to  $\text{ePAT}_{\text{tf}, \Sigma}$  if and only if, for every  $x_j \in \text{var}(\alpha)$ , there exists an  $w \in T_\alpha$  such that, for every substitution  $\sigma : \text{Pat}_{\text{tf}} \rightarrow \Sigma^*$  with  $\sigma(\alpha) = w$ , there is an  $A \in \Sigma$  with  $|\sigma(x_j)|_A = 1$  and  $|\sigma(\alpha)|_A = |\alpha|_{x_j}$ .

**Proof.** We begin the proof with the *if* part of the theorem. Hence, let  $T_\alpha = \{w_1, w_2, \dots, w_n\}$  be a set of words satisfying the above condition. Moreover, let  $\beta \in \text{Pat}_{\text{tf}}$  be any pattern such that  $T_\alpha \subseteq L_\Sigma(\beta) \subseteq L_\Sigma(\alpha)$ . We show that this assumption implies  $L_\Sigma(\beta) = L_\Sigma(\alpha)$ , and therefore  $\beta$  is not a passe-partout for  $\alpha$  and  $T_\alpha$ .

As  $L_\Sigma(\beta) \subseteq L_\Sigma(\alpha)$ , there is a morphism  $\phi : X^* \rightarrow X^*$  with  $\phi(\alpha) = \beta$  (cf. Theorem 2). Furthermore, due to  $T_\alpha \subseteq L_\Sigma(\beta)$ , there exists a set  $\{\sigma_1, \sigma_2, \dots, \sigma_n\}$  of substitutions with  $\sigma_i(\beta) = w_i$ ,  $1 \leq i \leq n$ . Note that, for every  $i$  with  $1 \leq i \leq n$ ,  $\sigma_i \circ \phi$  is a substitution which, when applied to  $\alpha$ , leads to  $w_i$ . Thus, and because of the fact that  $T_\alpha$  satisfies the condition of Theorem 10, for every  $x_j \in \text{var}(\alpha)$  there necessarily exist a  $j'$ ,  $1 \leq j' \leq n$  and an  $A \in \Sigma$  such that  $\sigma_{j'}(\phi(x_j)) = v_1 A v_2$ , where  $v_1, v_2 \in \Sigma \setminus \{A\}$ ; moreover,  $|\sigma_{j'}(\alpha)|_A = |\alpha|_{x_j}$ . Consequently, in order to generate this unique letter  $A$ ,  $\phi(x_j)$  must contain a unique variable, i.e., for some  $j'' \in \mathbb{N}$ ,  $\phi(x_j) = \gamma_1 x_{j''} \gamma_2$  with  $\gamma_1, \gamma_2 \in X^*$  and  $|\phi(\alpha)|_{x_{j''}} = |\alpha|_{x_j}$ . For every  $x_k \in \text{var}(\beta)$  and for every  $j''$  with  $x_j \in \text{var}(\alpha)$ , we now define the morphism  $\psi : X^* \rightarrow X^*$  by

$$\psi(x_k) := \begin{cases} x_j, & k = j'', \\ \varepsilon, & \text{else.} \end{cases}$$

As stated above, for every  $j''$ , the number of occurrences of  $x_{j''}$  in  $\beta$  equals the number of occurrences of  $x_j$  in  $\alpha$ . Furthermore, each  $x_{j''}$  solely occurs in the respective  $\phi(x_j)$ , and therefore the order of the variables  $x_{j''}$  in  $\beta$  corresponds to the order of the variables  $x_j$  in  $\alpha$ . Thus,  $\psi(\beta) = \alpha$ , which implies  $L_\Sigma(\beta) \supseteq L_\Sigma(\alpha)$  (according to Theorem 2). Together with the condition  $L_\Sigma(\alpha) \supseteq L_\Sigma(\beta)$ , this leads to  $L_\Sigma(\beta) = L_\Sigma(\alpha)$ . Hence, there is no passe-partout  $\beta \in \text{Pat}_{\text{tf}}$  for  $\alpha$  and  $T_\alpha$ .

Consequently,  $T_\alpha$  is a telltale for  $L_\Sigma(\alpha)$  with respect to  $\text{ePAT}_{\text{tf}, \Sigma}$ , which proves the *if* part of Theorem 10.

In order to prove the *only if* part of the theorem, we regard any finite set  $W \subseteq L_\Sigma(\alpha)$  which does not have the properties noted in Theorem 10. We show that this assumption implies that there exists a terminal-free passe-partout for  $\alpha$  and  $W$ , so that  $W$  is not a telltale for  $L_\Sigma(\alpha)$  with respect to  $\text{ePAT}_{\text{tf}, \Sigma}$ .

Hence, let  $W := \{w_1, w_2, \dots, w_n\} \subseteq L_\Sigma(\alpha)$ ,  $n \in \mathbb{N}$ , and let the alphabet  $\Sigma$  under consideration be specified by  $\Sigma := \{a_1, a_2, \dots, a_m\}$ ,  $m \geq 2$ , with  $a_l \neq a_{l'}$  for  $1 \leq l, l' \leq m$ ,  $l \neq l'$ . Furthermore, for every  $i$ ,  $1 \leq i \leq n$ , and for every  $l$ ,  $1 \leq l \leq m$ , let the inverse substitution  $\bar{\sigma}_i : \Sigma^* \rightarrow X^*$  be given by  $\bar{\sigma}_i(a_l) := x_{m(i-1)+l}$ . Below, we shall use these inverse substitutions  $\bar{\sigma}_i$  for constructing the said passe-partout.

As  $W$  does not satisfy the condition of Theorem 10 there must be a variable  $x_\# \in \text{var}(\alpha)$  such that, for every  $w_i \in W$ , there exists a substitution  $\sigma_{i,\#}$  with  $\sigma_{i,\#}(\alpha) = w_i$  and, for every letter  $A \in \Sigma$ ,  $|\sigma_{i,\#}(x_\#)|_A \neq 1$  or  $|\sigma_{i,\#}(\alpha)|_A \neq |\alpha|_{x_\#}$ . Using these substitutions  $\sigma_{i,\#}$ , we define for every  $x_j \in \text{var}(\alpha)$  a pattern  $\beta_j$  by  $\beta_j := \bar{\sigma}_1(\sigma_{1,\#}(x_j)) \bar{\sigma}_2(\sigma_{2,\#}(x_j)) [\dots] \bar{\sigma}_n(\sigma_{n,\#}(x_j))$ . We now introduce a morphism  $\phi : X^* \rightarrow X^*$  by, for every  $x_j \in \text{var}(\alpha)$ ,  $\phi(x_j) := \beta_j$ , and we denote  $\beta := \phi(\alpha)$ .

In order to conclude the proof, we have to show that  $\beta$  indeed is a passe-partout for  $\alpha$  and  $W$ , i.e.  $W \subseteq L_\Sigma(\beta) \subset L_\Sigma(\alpha)$ . The first aspect to be proven, namely  $W \subseteq L_\Sigma(\beta)$ , directly follows from the definition of  $\beta$ : For every  $w_i \in W$ , we can simply refer to the substitution  $\sigma'_i$  reading, for every  $x_k \in \text{var}(\beta)$ ,

$$\sigma'_i(x_k) := \begin{cases} a_{k-m(i-1)}, & 1 \leq k - m(i-1) \leq m, \\ \varepsilon, & \text{else,} \end{cases}$$

which yields  $\sigma'_i(\beta) = w_i$ .

With regard to the second necessary property of  $\beta$ , i.e.  $L_\Sigma(\beta) \subset L_\Sigma(\alpha)$ , we know – due to the construction of  $\beta$  which is based on the morphism  $\phi$  – that  $L_\Sigma(\beta) \subseteq L_\Sigma(\alpha)$  (according to Theorem 2). Thus, we merely need to prove that  $L_\Sigma(\alpha)$  is not a subset of  $L_\Sigma(\beta)$  or, equivalently, that there is no morphism mapping  $\beta$  onto  $\alpha$ . To this end, we assume to the contrary that there is such a morphism  $\psi : X^* \rightarrow X^*$  with  $\psi(\beta) = \alpha$ . Then, because of the succinctness of  $\alpha$ , the patterns  $\alpha, \beta$  and the morphisms  $\phi, \psi$  satisfy the conditions of Lemma 6, which states that, for every  $x_j \in \text{var}(\alpha)$ ,  $\psi(\phi(x_j)) = x_j$ . With regard to  $x_\# \in \text{var}(\alpha)$ , however, we know that, for every  $i$  with  $1 \leq i \leq n$  and for every letter  $A \in \Sigma$  occurring in  $\sigma_{i,\#}(x_\#)$ ,  $|\sigma_{i,\#}(\alpha)|_A > |\alpha|_{x_\#}$ . Hence, for every  $x_k \in \text{var}(\beta_\#)$ ,  $|\beta|_{x_k} > |\alpha|_{x_\#}$  and

therefore  $x_{\sharp} \notin \text{var}(\psi(x_k))$ . Consequently,  $\psi(\phi(x_{\sharp})) \neq x_{\sharp}$ , which evidently contradicts Lemma 6. Thus, there is no morphism  $\psi$  with  $\psi(\beta) = \alpha$ ; this implies  $L_{\Sigma}(\beta) \not\subseteq L_{\Sigma}(\alpha)$  and, referring to the existence of  $\phi$ ,  $L_{\Sigma}(\beta) \subset L_{\Sigma}(\alpha)$ .

Consequently,  $\beta$  is a passe-partout for  $\alpha$  and  $W$ , which proves the *only if* part of Theorem 10.  $\square$

Evidently, Theorem 10 allows us to discuss the problem of the learnability of the full class of terminal-free E-pattern languages in a manner that shows no immediate connections to the algorithmic definition of learning (cf. Section 2) anymore. It is based on the fundamental criterion by Angluin [2] (cf. Theorem 3) that introduces a language theoretical (or perhaps rather *topological*) analogue to the learnability of indexed families, and it replaces this view by an equivalent problem on combinatorics on morphisms. More precisely, Theorem 10 for each word in a given set examines all of its generating substitutions, and, hence, it deals with the ambiguity of words with respect to a fixed pattern (for recent insights into the existence of *unambiguous* words in terminal-free pattern languages, see Freydenberger et al. [7]). Thus, surprisingly, the (non-)learnability of  $\text{ePAT}_{\text{tf}}$  is manifestly related to the fields of *equality sets* of morphisms (and, thus, even to the Post Correspondence Problem; see, e.g., Harju, Karhumäki [10] and Lipponen, Păun [19]) and to *word equations* (see, e.g., Makanin [21]). Contrary to this, the connections between our subject and the studies on the *ambiguity of pattern languages* as conducted by Mateescu, Salomaa [22] are rather weak, since [22] asks for the existence of a bound  $n \in \mathbb{N}$  such that, for a fixed pattern  $\alpha$  and for *all* words  $w$  in  $L(\alpha)$ , there exist at most  $n$  different substitutions  $\sigma$  satisfying  $\sigma(\alpha) = w$ , whereas we are interested in the ambiguity of certain *selected* words, and we have to study the *shape* of their generating substitutions rather than the *number* of these substitutions.

We conclude this section with a number of examples which are mainly meant to illustrate Theorem 10. Additionally, we use them to provide some concrete – and more or less obvious – insights into telltales of terminal-free E-pattern languages. Our first example demonstrates that a singular set of words can be a telltale, even though Theorem 10 requires that *for every variable* in the pattern *there exists* a word in the telltale:

**Example 11.** Let  $\Sigma := \{a, b\}$ ,  $\alpha := x_1^2 x_2^2$ ,  $w := aabb$  and  $T_{\alpha} := \{w\}$ . Evidently,  $\alpha$  is succinct. Then there is exactly one substitution  $\sigma$  with  $\sigma(\alpha) = w$ , namely  $\sigma(x_1) = a$  and  $\sigma(x_2) = b$ . Thus – since  $|w|_a = 2 = |\alpha|_{x_1}$ ,  $|w|_b = 2 = |\alpha|_{x_2}$  and  $|\sigma(x_1)|_a = 1 = |\sigma(x_2)|_b$  – the singular set  $T_{\alpha}$  is a telltale for  $L_{\Sigma}(\alpha)$  with respect to  $\text{ePAT}_{\text{tf}, \Sigma}$ .

Of course, such an example is only possible if the pattern under consideration does not contain more variables as there are different letters in the corresponding alphabet. Contrary to this, Reidenbach [30] demonstrates that a major subclass of terminal-free E-pattern languages can be learned using telltales that contain the single word generated by the simple injective substitution  $\sigma$  given by  $\sigma(x_j) := ab^j$ ,  $j \in \mathbb{N}$ . However, if we examine the telltales of terminal-free E-pattern languages *with respect to*  $\text{ePAT}_{\text{tf}}$  – as done by Theorem 10 – then such an approach necessarily fails for all nontrivial cases:

**Example 12.** Let  $\Sigma := \{a, b\}$ , and let the substitution  $\sigma : \text{Pat}_{\text{tf}} \rightarrow \Sigma^*$  be given by  $\sigma(x_j) = ab^j$ ,  $j \in \mathbb{N}$ . Then, for every succinct pattern  $\alpha$  with  $|\text{var}(\alpha)| \geq 2$ , for every variable  $x_j \in \text{var}(\alpha)$  and for every letter  $A$  occurring in  $\sigma(\alpha)$ , it is  $|\sigma(\alpha)|_A > |\alpha|_{x_j}$ . Thus, for no such pattern, the singular set  $W_{\alpha} := \{\sigma(\alpha)\}$  is a telltale for  $L_{\Sigma}(\alpha)$  with respect to  $\text{ePAT}_{\text{tf}, \Sigma}$ .

Hence, the question of whether a given set of words can serve as a telltale for a terminal-free E-pattern language essentially depends on the concrete class of languages under consideration, and this also holds for nontrivial example classes. In this regard, Theorem 10 of course gives the most selective criterion as it determines the telltales with regard to the *full* class of terminal-free E-pattern languages; thus the criterion at least is sufficient for all classes of terminal-free E-pattern languages.

As pointed out above, Theorem 10 reveals that a word  $w$  is a useful element of a telltale for the language generated by an arbitrary succinct terminal-free pattern  $\alpha$  if and only if, first, there is a substitution  $\sigma$  satisfying  $\sigma(\alpha) = w$  and assigning a unique letter  $A$  to a variable in  $\alpha$  and, second, the ambiguity of  $w$  (with respect to  $\alpha$ , of course) is restricted in a particular manner. If we now consider the reasoning in Example 12 then we can observe that it solely refers to the fact that the injective substitution examined therein does not conform to the former of these requirements. Contrary to this, concerning the latter condition, [30] shows that, for a large class of patterns, the said substitution leads to words with the desirable property of unambiguity. Therefore, in the subsequent example we examine what happens if we modify  $\sigma$  such that it largely keeps its original structure, but nevertheless assigns a unique letter to a variable:

**Example 13.** Let  $\Sigma := \{a, b, c\}$ , and let the substitutions  $\sigma_i : \text{Pat}_{\text{tf}} \longrightarrow \Sigma^*$  be given by

$$\sigma_i(x_j) := \begin{cases} c, & i = j, \\ ab^j, & \text{else,} \end{cases}$$

$i, j \in \mathbb{N}$ . Let the succinct pattern  $\alpha$  be given by  $\alpha := x_1^2 x_2^2 x_3^2$ . Note that if we restrict ourselves to a binary alphabet then – according to Reidenbach [30] – the language of  $\alpha$  does not have a telltale with respect to  $\text{ePAT}_{\text{tf}, \{a, b\}}$  (so that, as mentioned in Section 1 and to be further discussed in Section 4,  $\text{ePAT}_{\text{tf}, \Sigma}$  is not learnable for  $|\Sigma| = 2$ ). Let

$$\begin{aligned} T_\alpha &:= \{\sigma_1(\alpha), \sigma_2(\alpha), \sigma_3(\alpha)\} \\ &= \{ccabbabbabbabb, ababccabbabb, abababbabbcc\}. \end{aligned}$$

Then it can be verified with a bit of effort (or, alternatively, it can be indirectly derived from an argument given by [30] on a set of patterns comprising  $\alpha$ ) that  $\sigma_1(\alpha)$ ,  $\sigma_2(\alpha)$  and  $\sigma_3(\alpha)$  are unambiguous with respect to  $\alpha$ . Thus, for every  $x_j \in \text{var}(\alpha)$  there exists a  $w \in T_\alpha$  (namely  $w = \sigma_j(\alpha)$ ) such that, for every substitution  $\sigma$  with  $\sigma(\alpha) = w$  (since  $w$  is unambiguous, there is only one such substitution, namely  $\sigma_j$ ) there exists an  $A \in \Sigma$  (evidently, it is  $A = c$ ) with  $|\sigma(x_j)|_A = 1$  and  $|\sigma(\alpha)|_A = |\alpha|_{x_j}$ . Consequently,  $T_\alpha$  is a telltale for  $L_\Sigma(\alpha)$  with respect to  $\text{ePAT}_{\text{tf}, \Sigma}$ .

Thus, as soon as a third letter is available in the alphabet, we can modify the substitution  $\sigma$  given by  $\sigma(x_j) := ab^j$ ,  $j \in \mathbb{N}$ , so that, concerning the prominent example pattern  $\alpha = x_1^2 x_2^2 x_3^2$ , the resulting substitutions  $\sigma_i$  introduced in Example 13 generate a telltale for  $L_\Sigma(\alpha)$  with respect to  $\text{ePAT}_{\text{tf}, \Sigma}$ ,  $|\Sigma| \geq 3$ . As explained above, the unambiguity of the  $\sigma_i(\alpha)$  contributes significantly to this desirable result.

With regard to other terminal-free patterns  $\alpha$ , however, the words  $\sigma_i(\alpha)$ ,  $i \in \mathbb{N}$ , are ambiguous, and among the languages generated by these patterns we can even find examples for which the substitutions under consideration do not lead to a telltale:

**Example 14.** Let the substitutions  $\sigma_i$  be given in accordance with Example 13, and let

$$\begin{aligned} \alpha &:= x_1 x_2 x_1 x_2 x_3 x_4 x_3 x_5 x_4 x_6 x_7 x_8 x_1 x_8 x_7 x_9 x_{10} x_{11} x_5 x_{11} x_5 \\ &\quad x_{12} x_6 x_{13} x_{12} x_6 x_{13} x_{14} x_9 x_{14} x_9 x_{15} x_{10} x_{15} x_{10}. \end{aligned}$$

By Theorem 7, it can be straightforwardly verified that  $\alpha$  is succinct. We now can demonstrate that  $W_\alpha := \{\sigma_j(\alpha) \mid x_j \in \text{var}(\alpha)\}$  is not a telltale for  $L_\Sigma(\alpha)$  by introducing a substitution  $\tau_3$  by  $\tau_3(x_1) := abab^2$ ,  $\tau_3(x_2) := \tau_3(x_3) := \tau_3(x_5) := \varepsilon$ ,  $\tau_3(x_4) := cab^4$ ,  $\tau_3(x_6) := bab^3$ ,  $\tau_3(x_7) := ba$ ,  $\tau_3(x_8) := b^6 ab^7 ab^8$ ,  $\tau_3(x_9) := b$ ,  $\tau_3(x_{10}) := b^9$ ,  $\tau_3(x_{11}) := ab^{11} ab^5$ ,  $\tau_3(x_{12}) := ab^{11}$ ,  $\tau_3(x_{13}) := b^3 ab^{13}$ ,  $\tau_3(x_{14}) := ab^{14} ab^8$ ,  $\tau_3(x_{15}) := ab^{15} ab$ . Then  $\tau_3(\alpha) = \sigma_3(\alpha)$ , but  $\tau_3$  maps  $x_3$  onto the empty word, so that it does not contain any unique letter  $A$  (as referred to in Theorem 10). Since, for every  $i$  satisfying  $i \neq 3$ , the substitution  $\sigma_i$  does not assign such a letter to  $x_3$ , either,  $W_\alpha$  indeed is not a telltale for  $L_\Sigma(\alpha)$  with respect to  $\text{ePAT}_{\text{tf}, \Sigma}$ ,  $|\Sigma| \geq 3$ . Thus, the substitutions  $\sigma_i$  introduced above do not lead to a telltale in general.

Consequently, if we wish to examine the learnability of  $\text{ePAT}_{\text{tf}}$  for some alphabet by Theorem 10 then we need to find more appropriate and probably more sophisticated candidates than the substitutions  $\sigma_i$  given in Example 13 – or we have to disprove their existence.

Still – apart from this overall learning theoretical goal to be tackled in Section 4 which naturally focuses on well-chosen types of substitutions – Theorem 10 can be used to examine arbitrary sets of words on the telltale property (with respect to any fixed terminal-free E-pattern language). Thus, it can reveal that telltales do not necessarily need to follow the rigid principles implemented by the substitutions given in Examples 11–14:

**Example 15.** Finally let  $\Sigma := \{a, b\}$  again. Let  $\alpha := x_1^3 x_2^2 x_3^4$ . It can be easily verified that  $\alpha$  is succinct. Then  $T_\alpha := \{(ab)^3 b^4, (ab)^2, a^5 b^4\}$  is a telltale (though by no means an optimal, i.e. shortest, one) for  $L_\Sigma(\alpha)$  with respect to  $\text{ePAT}_{\text{tf}, \Sigma}$ : For every substitution  $\sigma_1$  generating the first word  $(ab)^3 b^4$ ,  $\sigma_1(x_1)$  contains the letter  $a$  with  $|\sigma_1(x_1)|_a = 1$  and  $|\sigma_1(\alpha)|_a = 3 = |\alpha|_{x_1}$ . The second word  $(ab)^2$  can only be generated by the substitution  $\sigma_2$  with  $\sigma_2(x_2) = ab$ ,  $\sigma_2(x_1) = \sigma_2(x_3) = \varepsilon$ , and therefore both  $a$  and  $b$  can serve as the unique letter with respect to  $x_2$ . The third word  $a^5 b^4$  again is unambiguous with respect to  $\alpha$ , and its only generating substitution maps  $x_3$  onto the unique letter  $b$ .

By [Theorems 7](#) and [10](#) we have two powerful combinatorial tools for dealing with terminal-free E-pattern languages over alphabets with at least two distinct letters. In the subsequent section we shall apply them to the problem of the learnability of  $\text{ePAT}_{\text{tf}, \Sigma}$  for  $|\Sigma| \geq 3$ .

#### 4. Inductive inference of terminal-free E-pattern languages

As mentioned in the context of [Example 13](#), it is shown by Reidenbach [[30](#)] that, for the pattern  $\alpha := x_1^2 x_2^2 x_3^2$  and for every finite  $W \subseteq L_{\Sigma}(\alpha)$ , there exists a terminal-free passe-partout if a binary terminal alphabet  $\Sigma$  is considered. Thus, the class of terminal-free E-pattern languages over such an alphabet is not learnable:

**Theorem 16** (Reidenbach [[30](#)]). *Let  $\Sigma$  be an alphabet,  $|\Sigma| = 2$ . Then  $\text{ePAT}_{\text{tf}, \Sigma}$  is not inferrable from positive data.*

As an immediate consequence thereof it can be concluded that the full class of E-pattern languages is not learnable, either, in that case.

Intuitively, [Theorem 16](#) demonstrates that the expressive power of substitutions (which are ordinary morphisms as long as we restrict ourselves to terminal-free patterns) mapping a pattern onto a word over a binary alphabet is not sufficient for generating morphic images that allow us to draw unequivocal conclusions about their common preimage. Hence, very roughly speaking, if  $|\Sigma| \geq 2$  then we cannot “encode” the structure of a terminal-free pattern  $\alpha$  into a finite sublanguage of  $L_{\Sigma}(\alpha)$ . Consequently, basic insights in the theory of codes, which say that a code over a binary alphabet is as powerful as a code over any larger alphabet, suggest that the negative result on binary alphabets might be extendable to all finite non-unary alphabets.

Contrary to these considerations, however, a closer look at the characterisation of telltales for terminal-free E-pattern languages as presented in [Theorem 10](#) and, in particular, as applied in [Example 13](#) demonstrates that there are E-pattern languages over a binary alphabet  $\Sigma$  for which there exists no telltale (with respect to  $\text{ePAT}_{\text{tf}, \Sigma}$ ), but, for the language over a ternary alphabet generated by the same pattern, there is a telltale. Our main result of the present section shows that, surprisingly, we can find such telltales for *all* terminal-free E-pattern languages over at least three distinct letters:

**Theorem 17.** *Let  $\Sigma$  be a finite alphabet,  $|\Sigma| \geq 3$ . Then  $\text{ePAT}_{\text{tf}, \Sigma}$  is inferrable from positive data.*

The proof for [Theorem 17](#) is given in [Section 4.1](#).

By [Theorem 17](#), we have answered the question of the learnability of  $\text{ePAT}_{\text{tf}, \Sigma}$  for all alphabet sizes:

**Corollary 18.** *Let  $\Sigma$  be an alphabet. Then  $\text{ePAT}_{\text{tf}, \Sigma}$  is inferrable from positive data if and only if  $|\Sigma| \neq 2$ .*

**Proof.** With regard to  $|\Sigma| = 1$  and  $|\Sigma| = \infty$ , the proof is given by Mitchell [[24](#)]. For  $|\Sigma| = 2$ , see [Theorem 16](#) and, for any other alphabet size, [Theorem 17](#).  $\square$

Consequently, [Corollary 18](#) demonstrates that the learnability of terminal-free E-pattern languages is discontinuous subject to the size of the terminal alphabet, which – referring to [Theorem 3](#) – necessarily implies that some fundamental intrinsic properties of  $\text{ePAT}_{\text{tf}, \Sigma}$  change under the alphabet extension from  $|\Sigma| = 2$  to  $|\Sigma| = 3$ . Nevertheless, we do not think that the corresponding phenomena are perfectly understood so far. In particular, it is noteworthy that the varying learnability results for  $\text{ePAT}_{\text{tf}, \Sigma}$  contrast with the continuous behaviour of the equivalence of terminal-free E-pattern languages over the alphabet sizes under consideration: as indirectly shown by [Theorem 2](#), two terminal-free patterns generate the same language over a binary alphabet if and only if they generate the same language over any alphabet with more than two letters.

Hence, in spite of the proof for [Theorem 17](#) to be given in the subsequent section, we still consider [Corollary 18](#) to be rather counter-intuitive. This is mainly caused by the observation that, on the one hand, our results essentially depend on the (non-)existence of morphisms sufficiently reflecting the structure of a pattern (see [Theorem 10](#)) and, on the other hand, that the stated properties do not at all match with the most elementary insights in coding theory. Therefore we expect that a deeper understanding of [Corollary 18](#) requires a further analysis of the special characteristics of morphisms in “combinatorial” contexts (as provided by pattern languages).

#### 4.1. Proof of Theorem 17

We begin our proof for Theorem 17 with the definition of the substitutions which, when applied to any succinct pattern  $\alpha$ , lead to a telltale for  $L_{\Sigma}(\alpha)$  with respect to  $\text{ePAT}_{\text{tf}, \Sigma}$ ; to this end we have to assume that  $\Sigma \supseteq \{a, b, c\}$ .

**Definition 19.** Let  $\Sigma$  be an alphabet,  $\{a, b, c\} \subseteq \Sigma$ . Then, for every  $i, j \in \mathbb{N}$ , the substitution  $\sigma_{\text{tf-tt}, i} : \text{Pat}_{\text{tf}} \longrightarrow \Sigma^*$  is given by

$$\sigma_{\text{tf-tt}, i}(x_j) := \begin{cases} \text{ab}^{3j-2}\text{a} \text{c} \text{ab}^{3j-1}\text{a} \text{ab}^{3j}\text{a}, & i = j, \\ \text{ab}^{3j-2}\text{a} \text{ab}^{3j-1}\text{a} \text{ab}^{3j}\text{a}, & \text{else.} \end{cases}$$

It can be immediately seen that, for every  $j$  with  $x_j \in \text{var}(\alpha)$ , the morphism  $\sigma_{\text{tf-tt}, j}$  maps the variable  $x_j$  onto a word containing the unique letter  $A = c$  referred to in Theorem 10. Hence, we merely have to show that every substitution  $\tau$  with  $\tau(\alpha) = \sigma_{\text{tf-tt}, j}(\alpha)$  also has this property, i.e.  $\tau(x_j) = \dots c \dots$ . Then Theorem 17 follows directly from Theorems 3 and 10. Unfortunately, however, the proof of the said property of the morphisms  $\tau$  is rather cumbersome.

For our proof of the suchlike “restricted” ambiguity of the  $\sigma_{\text{tf-tt}, i}(\alpha)$  (and, as to be shown in Section 4.2, also for the telltale property of these words), it is essential that  $\sigma_{\text{tf-tt}, i}$  maps each variable onto a word that consists of three segments  $\text{ab}^m\text{a}$ ,  $m \in \mathbb{N}$ . Since each of these segments in  $\sigma_{\text{tf-tt}, i}(\alpha)$  solely is generated by the occurrences of some particular  $x_j$  in  $\alpha$ , we can unequivocally call each word  $\text{ab}^{3j-p}\text{a}$ ,  $p \in \{0, 1, 2\}$ , a *segment of  $\sigma_{\text{tf-tt}, i}(x_j)$* . In order to address the segments more precisely we henceforth additionally use the following terms: For any variable  $x_j \in X$ , we call  $\text{ab}^{3j-2}\text{a}$  the *left segment*,  $\text{ab}^{3j-1}\text{a}$  the *inner segment* and  $\text{ab}^{3j}\text{a}$  the *right segment* of  $\sigma_{\text{tf-tt}, i}(x_j)$ . Furthermore, we extend this terminology to segments of  $\sigma_{\text{tf-tt}, i}(\gamma)$  for any (sub)pattern  $\gamma \in \text{Pat}_{\text{tf}}$ : A word  $w \in \{a, b, c\}^+$  is said to be a segment of  $\sigma_{\text{tf-tt}, i}(\gamma)$  if and only if there exists a variable  $x_j \in \text{var}(\gamma)$  such that  $w$  is a segment of  $\sigma_{\text{tf-tt}, i}(x_j)$ .

In fact, our subsequent Lemmata 20–24 are very similar to analogous lemmata given by Freydenberger et al. [7] on a different substitution  $\sigma_{\beta}^{\text{su}}$  which also maps every variable onto a word that consists of three distinct segments, and a major part of both lines of reasoning is even verbatim the same. For the sake of a self-contained presentation and since it requires some effort to adapt  $\sigma_{\beta}^{\text{su}}$  and the notations used in the mentioned paper to the needs of the present section, we nevertheless give all technical details.

Before we present the crucial Lemma 24, we formulate four lemmata which feature simple combinatorial observations on those variables  $x_j$  in  $\alpha$  for which – with regard to any substitution  $\tau$  with  $\tau(\alpha) = \sigma_{\text{tf-tt}, i}(\alpha)$  – the word  $\tau(x_j)$  contains any segment of  $\sigma_{\text{tf-tt}, i}(x_j)$  (note that a straightforward reasoning proves the existence of such variables even in prolix patterns). Since these lemmata are needed for an unobstructed understanding of our main argumentation in Lemma 24, it is important to keep them in mind.

According to our first observation, for every  $i \in \mathbb{N}$ , for every substitution  $\tau$  with  $\tau(\alpha) = \sigma_{\text{tf-tt}, i}(\alpha)$  and for every variable  $x_j$  in  $\alpha$ ,  $\tau(x_j)$  contains any complete segment of  $\sigma_{\text{tf-tt}, i}(x_j)$  at most once:

**Lemma 20.** Let  $\alpha, \beta \in \text{Pat}_{\text{tf}}$ , and let  $i \in \mathbb{N}$ . Let  $\tau : \text{Pat}_{\text{tf}} \longrightarrow \{a, b, c\}^*$  be any substitution with  $\tau(\alpha) = \sigma_{\text{tf-tt}, i}(\alpha)$ . Then, for every  $x_j \in \text{var}(\alpha)$  and for every  $p \in \{0, 1, 2\}$ ,  $\tau(x_j) \neq \dots \text{ab}^{3j-p}\text{a} \dots \text{ab}^{3j-p}\text{a} \dots$ .

**Proof.** Assume to the contrary that there is a variable  $x_j \in \text{var}(\alpha)$  with  $|\tau(x_j)|_{\text{ab}^{3j-p}\text{a}} \geq 2$ . Then  $|\tau(\alpha)|_{\text{ab}^{3j-p}\text{a}} \geq 2|\alpha|_{x_j} > |\alpha|_{x_j} = |\sigma_{\text{tf-tt}, i}(\alpha)|_{\text{ab}^{3j-p}\text{a}}$ . This contradicts the condition  $\tau(\alpha) = \sigma_{\text{tf-tt}, i}(\alpha)$ .  $\square$

The next lemma says that if, for any substitution  $\tau$  with  $\tau(\alpha) = \sigma_{\text{tf-tt}, i}(\alpha)$  and for any variable  $x_j \in \text{var}(\alpha)$ ,  $\tau(x_j)$  contains the left and the inner segments (or the inner and the right segments) of  $\sigma_{\text{tf-tt}, i}(x_j)$  then these segments occur in the “natural” order (i.e. in the order dictated by  $\sigma_{\text{tf-tt}, i}$ ):

**Lemma 21.** Let  $\alpha, \beta \in \text{Pat}_{\text{tf}}$ , and let  $i \in \mathbb{N}$ . Let  $\tau : \text{Pat}_{\text{tf}} \longrightarrow \{a, b, c\}^*$  be any substitution with  $\tau(\alpha) = \sigma_{\text{tf-tt}, i}(\alpha)$ . For every  $x_j \in \text{var}(\alpha)$  and for every  $p \in \{1, 2\}$ , if  $\tau(x_j) = \dots \text{ab}^{3j-p}\text{a} \dots$  and  $\tau(x_j) = \dots \text{ab}^{3j-p+1}\text{a} \dots$  then, for some  $v \in \{e, c\}$ ,  $\tau(x_j) = \dots \text{ab}^{3j-p}\text{a} v \text{ab}^{3j-p+1}\text{a} \dots$ .

**Proof.** Because of  $\tau(\alpha) = \sigma(\alpha)$ , we have  $|\tau(\alpha)|_{\text{ab}^{3j-p}\text{a}} = |\tau(\alpha)|_{\text{ab}^{3j-p+1}\text{a}} = |\tau(\alpha)|_{\text{ab}^{3j-p}\text{a} v \text{ab}^{3j-p+1}\text{a}} = |\alpha|_j$ . Thus, for every occurrence of  $\text{ab}^{3j-p}\text{a}$  and of  $\text{ab}^{3j-p+1}\text{a}$  in  $\tau(\alpha)$ ,  $\tau(\alpha) = \dots \text{ab}^{3j-p}\text{a} v \text{ab}^{3j-p+1}\text{a} \dots$ . Therefore, our conditions  $\tau(x_j) = \dots \text{ab}^{3j-p}\text{a} \dots$  and  $\tau(x_j) = \dots \text{ab}^{3j-p+1}\text{a} \dots$  imply  $\tau(x_j) = \dots \text{ab}^{3j-p}\text{a} v \text{ab}^{3j-p+1}\text{a} \dots$ .  $\square$



By a similar reasoning we can conclude that if, for any substitution  $\tau$  with  $\tau(\alpha) = \sigma_{\text{tf-tt},i}(\alpha)$  and for any variable  $x_j$  in  $\alpha$ ,  $\tau(x_j)$  contains the left and the right segments of  $\sigma_{\text{tf-tt},i}(x_j)$  then it must also contain the inner segment of  $\sigma_{\text{tf-tt},i}(x_j)$  and, again, these segments must occur in the canonical order:

**Lemma 22.** *Let  $\alpha, \beta \in \text{Pat}_{\text{tf}}$ , and let  $i \in \mathbb{N}$ . Let  $\tau : \text{Pat}_{\text{tf}} \rightarrow \{a, b, c\}^*$  be any substitution with  $\tau(\alpha) = \sigma_{\text{tf-tt},i}(\alpha)$ . For every  $x_j \in \text{var}(\alpha)$ , if  $\tau(x_j) = \dots ab^{3j-2}a \dots$  and  $\tau(x_j) = \dots ab^{3j}a \dots$  then, for some  $v \in \{\varepsilon, c\}$ ,  $\tau(x_j) = \dots ab^{3j-2}a v ab^{3j-1}a ab^{3j}a \dots$ .*

We conclude our list of basic properties of any  $x_j \in \text{var}(\alpha)$  for which  $\tau(x_j)$  contains a segment  $s = ab^{3j-p}a$ ,  $p \in \{0, 1, 2\}$ , with an immediate consequence of Lemma 20. Since  $\tau(x_j)$  contains this segment  $s$  only once, we reliably know that, for any  $n$ ,  $1 \leq n \leq |\alpha|_j$ , the  $n$ th occurrence (counted from the left) of  $x_j$  in  $\alpha$  under both  $\sigma$  and  $\tau$  necessarily generates the  $n$ th occurrence of  $s$  in  $\sigma_{\text{tf-tt},i}(\alpha) = \tau(\alpha)$ . Thus, from the said feature of  $\tau(x_j)$  we may not only draw additional “local” conclusions on  $\tau(x_j)$  as demonstrated by Lemmas 20–22, but also “global” ones (which – admittedly – are rather weak) on  $\tau(x_{j'})$  for every  $x_{j'} \in \text{var}(\alpha)$ . In anticipation of the requirements of the subsequent main Lemma 24 and for the sake of a more concise presentation, we focus on a variable  $x_j \in \text{var}(\alpha)$  for which  $\tau(x_j)$  contains the inner segment of  $\sigma_{\text{tf-tt},i}(x_j)$ :

**Lemma 23.** *Let  $\alpha, \beta \in \text{Pat}_{\text{tf}}$ . Let, for some variable  $x_j \in \text{var}(\alpha)$  and  $\alpha_1, \alpha_2 \in X^*$ ,  $\alpha = \alpha_1 x_j \alpha_2$ . Let  $i \in \mathbb{N}$  and  $\tau : \text{Pat}_{\text{tf}} \rightarrow \{a, b, c\}^*$  be any substitution with  $\tau(\alpha) = \sigma_{\text{tf-tt},i}(\alpha)$ . If, for some  $w_1, w_2 \in \{a, b, c\}^*$ ,  $\tau(x_j) = w_1 ab^{3j-1}a w_2$  then*

- $\tau(\alpha_1) w_1 = \sigma(\alpha_1) ab^{3j-2}a$  and
- $w_2 \tau(\alpha_2) = ab^{3j}a \sigma(\alpha_2)$ .

We now proceed with our main lemma which says that, for every  $i \in \mathbb{N}$ , for every succinct pattern  $\alpha$ , for every substitution  $\tau$  with  $\tau(\alpha) = \sigma_{\text{tf-tt},i}(\alpha)$  and for every  $x_j \in \text{var}(\alpha)$ ,  $\tau(x_j)$  contains at least the complete inner segment, the rightmost letter of left segment and the leftmost letter of the right segment of  $\sigma_{\text{tf-tt},i}(x_j)$ . Thus, it follows immediately, that, for every substitution  $\tau$  with  $\tau(\alpha) = \sigma_{\text{tf-tt},j}(\alpha)$ ,  $\tau(x_j)$  contains exactly one occurrence of the letter  $c$  which is inserted by  $\sigma_{\text{tf-tt},j}$  between the left and the inner segments of  $\sigma_{\text{tf-tt},j}(x_j)$ . This implies that, for every succinct pattern  $\alpha$ , the set  $T_\alpha := \{\sigma_{\text{tf-tt},j}(\alpha) \mid x_j \in \text{var}(\alpha)\}$  satisfies the conditions of Theorem 10 and, thus, that  $T_\alpha$  is a telltale for  $L_\Sigma(\alpha)$  with respect to every  $\Sigma$  containing at least three distinct letters. Note that the subsequent lemma can be easily extended such that it characterises succinctness (cf. Freydenberger et al. [7]). With regard to our needs, however, this fact is of no importance, and therefore we omit the corresponding statement.

**Lemma 24.** *Let  $\alpha \in \text{Pat}_{\text{tf}}$  be a succinct pattern. Then, for every  $i \in \mathbb{N}$ , for every  $x_j \in \text{var}(\alpha)$  and for every substitution  $\tau : \text{Pat}_{\text{tf}} \rightarrow \{a, b, c\}^*$  with  $\tau(\alpha) = \sigma_{\text{tf-tt},i}(\alpha)$ ,*

$$\tau(x_j) = \begin{cases} \dots a c ab^{3j-1}a a \dots, & i = j, \\ \dots a ab^{3j-1}a a \dots, & \text{else.} \end{cases}$$

**Proof.** We argue by contraposition. Consequently, we show that if there exists a substitution  $\tau$  with  $\tau(\alpha) = \sigma_{\text{tf-tt},i}(\alpha)$  and, for some  $x_j \in \text{var}(\alpha)$ ,  $\tau(x_j) \neq \dots a v ab^{3j-1}a a \dots$ ,  $v \in \{\varepsilon, c\}$ , then  $\alpha$  is prolix:

We start with a partition of  $\text{var}(\alpha)$  into subsets  $X_1, X_2, X_3$  depending on any substitution  $\tau$  satisfying the said conditions. From an informal point of view, this partition is given as follows: First, let  $X_1$  be the set of all variables  $x_j$  in  $\alpha$  such that  $\tau(x_j)$  contains the inner segment of  $\sigma_{\text{tf-tt},i}(x_j)$ , at least one letter of the left segment of  $\sigma_{\text{tf-tt},i}(x_j)$ , at least one letter of the right segment of  $\sigma_{\text{tf-tt},i}(x_j)$  and at least one complete segment of some  $\sigma_{\text{tf-tt},i}(x_{j'})$ ,  $j' \neq j$ . Second, let  $X_2$  be the set of all variables  $x_j$  in  $\alpha$  such that  $\tau(x_j)$  does not contain any letter of at least one segment of  $\sigma_{\text{tf-tt},i}(x_j)$ . Third (and last), let  $X_3$  be the set of all variables  $x_j$  in  $\alpha$  such that  $\tau(x_j)$  contains the inner segment of  $\sigma_{\text{tf-tt},i}(x_j)$ , at least one letter of the left segment of  $\sigma_{\text{tf-tt},i}(x_j)$ , at least one letter of the right segment of  $\sigma_{\text{tf-tt},i}(x_j)$ , but no complete segment of some  $\sigma_{\text{tf-tt},i}(x_{j'})$ ,  $j' \neq j$ .

Since  $\tau(\alpha) = \sigma_{\text{tf-tt},i}(\alpha)$  and thus, for every  $x_j \in \text{var}(\alpha)$ ,  $\tau(x_j)$  is a subword of  $\sigma_{\text{tf-tt},i}(\alpha)$  this vague definition of  $X_1, X_2$  and  $X_3$  results in several evident restrictions on the images under  $\tau$  of the variables in  $\alpha$  (cf. Lemmata 20–23)

such that the introduced subsets of  $\text{var}(\alpha)$  read formally:

$$\begin{aligned} X_1 &:= \{x_j \in \text{var}(\alpha) \mid \tau(x_j) = \dots \text{ab}^{3j'} \text{a} \text{ab}^{3j-2} \text{a} v \text{ab}^{3j-1} \text{a} \text{a} \dots \text{ or} \\ &\quad \tau(x_j) = \dots \text{a} v \text{ab}^{3j-1} \text{a} \text{ab}^{3j} \text{a} \text{ab}^{3j'-2} \text{a} \dots, \\ &\quad x_{j'} \in \text{var}(\alpha), v \in \{\varepsilon, c\}\}, \\ X_2 &:= \{x_j \in \text{var}(\alpha) \mid \tau(x_j) \neq \dots \text{a} v \text{ab}^{3j-1} \text{a} \text{a} \dots, v \in \{\varepsilon, c\}\}, \\ X_3 &:= \{x_j \in \text{var}(\alpha) \mid \tau(x_j) = \dots \text{a} v \text{ab}^{3j-1} \text{a} \text{a} \dots \text{ and} \\ &\quad \tau(x_j) \neq \dots \text{ab}^{3j'} \text{a} \text{ab}^{3j-2} \text{a} \dots \text{ and} \\ &\quad \tau(x_j) \neq \dots \text{ab}^{3j} \text{a} \text{ab}^{3j'-2} \text{a} \dots, \\ &\quad x_{j'} \in \text{var}(\alpha), v \in \{\varepsilon, c\}\}. \end{aligned}$$

Directly from the definition, it can be verified that  $X_1 \cap X_2 = X_1 \cap X_3 = X_2 \cap X_3 = \emptyset$  and  $X_1 \cup X_2 \cup X_3 = \text{var}(\alpha)$ . According to our initial condition, there is a variable  $x_j \in \text{var}(\alpha)$  with  $\tau(x_j) \neq \dots \text{a} v \text{ab}^{3j-1} \text{a} \text{a} \dots, v \in \{\varepsilon, c\}$ , and therefore  $X_2 \neq \emptyset$ . Note that our subsequent argumentation in Claim 3 shows that this leads to  $X_1 \neq \emptyset$ .

As we now wish to show that  $X_2 \neq \emptyset$  implies  $\alpha$  being prolix, we need to find an appropriate decomposition of  $\alpha$  satisfying the three conditions of Theorem 7. We start our argumentation with the following one:

$$\alpha = \bar{\beta}_0 \bar{\gamma}_1 \bar{\beta}_1 \bar{\gamma}_2 \bar{\beta}_2 [\dots] \bar{\beta}_{\bar{m}-1} \bar{\gamma}_{\bar{m}} \bar{\beta}_{\bar{m}}$$

with  $\bar{m} \geq 1$  and

- $\bar{\beta}_0, \bar{\beta}_{\bar{m}} \in X_3^*$  and  $\bar{\beta}_k \in X_3^+, 1 \leq k \leq \bar{m} - 1$ , and
- $\bar{\gamma}_k \in (X_1 \cup X_2)^+, 1 \leq k \leq \bar{m}$ .

Note that  $\bar{m} \geq 1$  is guaranteed because of  $X_2 \neq \emptyset$ .

Obviously, this decomposition is unique, and it satisfies condition (ii) of Theorem 7 since  $X_1, X_2$  and  $X_3$  are disjoint:

*Claim 1. For every  $k, k', 1 \leq k, k' \leq \bar{m}$ ,  $\text{var}(\bar{\gamma}_k) \cap \text{var}(\bar{\beta}_{k'}) = \emptyset$ .*

Concerning condition (i) of Theorem 7 we need to examine the given decomposition of  $\alpha$  in a bit more detail. The subsequent claim says that, for every  $\bar{\gamma}_k$ ,  $\tau(\bar{\gamma}_k)$  “almost” corresponds to  $\sigma_{\text{tf-tt},i}(\bar{\gamma}_k)$ , i.e.  $\tau(\bar{\gamma}_k)$  contains *at least*  $3|\bar{\gamma}_k| - 2$  complete segments of  $\sigma_{\text{tf-tt},i}(\bar{\gamma}_k)$  (and potentially some letters of two other segments) and *at most*  $3|\bar{\gamma}_k|$  complete segments of  $\sigma_{\text{tf-tt},i}(\bar{\gamma}_k)$  and, moreover, for every variable  $x_j \in \text{var}(\bar{\gamma}_k)$  the inner segment of  $\sigma_{\text{tf-tt},i}(x_j)$  as often as  $x_j$  is contained in  $\bar{\gamma}_k$ , and it does not contain any segment of  $\sigma_{\text{tf-tt},i}(x_{j'})$  if  $x_{j'} \notin \text{var}(\bar{\gamma}_k)$ :

*Claim 2. For every  $\bar{\gamma}_k = x_{j_1} x_{j_2} [\dots] x_{j_s}$ ,  $s \in \mathbb{N}$ , for every  $x_{j'} \in \text{var}(\alpha)$  and for some  $v_1, v_2, \dots, v_s \in \{\varepsilon, c\}$*

$$\begin{aligned} \tau(\bar{\gamma}_k) &= \dots \text{a} v_1 \text{ab}^{3j_1-1} \text{a} \text{ab}^{3j_1} \text{a} \text{ab}^{3j_2-2} \text{a} [\dots] v_s \text{ab}^{3j_s-1} \text{a} \text{a} \dots \text{ and} \\ \tau(\bar{\gamma}_k) &\neq \dots \text{a} v_1 \text{ab}^{3j_1-1} \text{a} \text{ab}^{3j_1} \text{a} [\dots] \text{ab}^{3j_s} \text{a} \text{ab}^{3j'-2} \text{a} \dots \text{ and} \\ \tau(\bar{\gamma}_k) &\neq \dots \text{ab}^{3j'} \text{a} \text{ab}^{3j_1-2} \text{a} v_1 \text{ab}^{3j_1-1} \text{a} [\dots] v_s \text{ab}^{3j_s-1} \text{a} \text{a} \dots \end{aligned}$$

For any subpattern  $\delta$  of  $\alpha$  satisfying the statement in Claim 2 we say that  $\tau(\delta)$  *corresponds to*  $\sigma_{\text{tf-tt},i}(\delta)$  (apart from a negligible prefix and suffix).

Claim 2 follows from the fact that every  $\bar{\gamma}_k$  is surrounded by  $\bar{\beta}_{k-1} \in X_3^*$  and  $\bar{\beta}_k \in X_3^*$ . Thus, with Lemma 23 applied to the variables in  $X_3$ ,  $\tau(\bar{\gamma}_k)$  is fixed by  $\tau(\bar{\beta}_{k-1})$  and  $\tau(\bar{\beta}_k)$ : as these two subwords of  $\tau(\alpha)$  by definition correspond to  $\sigma_{\text{tf-tt},i}(\bar{\beta}_{k-1})$  and  $\sigma_{\text{tf-tt},i}(\bar{\beta}_k)$ , respectively,  $\tau(\bar{\gamma}_k)$  must also correspond to  $\sigma_{\text{tf-tt},i}(\bar{\gamma}_k)$ . Consequently – and since by definition, for no  $\delta \in X_1^+$ ,  $\tau(\delta)$  corresponds to  $\sigma_{\text{tf-tt},i}(\delta) - \bar{\gamma}_k \notin X_1^+$ .

We proceed our argumentation on condition (i) of Theorem 7 being satisfied for the regarded decomposition by a closer look at the images under  $\tau$  of those subpatterns  $\delta$  of  $\alpha$  which exclusively consist of variables in  $X_2$ . In this regard we can see that  $\tau(\delta)$  necessarily does not correspond to  $\sigma_{\text{tf-tt},i}(\delta)$ :

*Claim 3. For every  $\delta = x_{j_1} x_{j_2} [\dots] x_{j_t}$  with  $t \in \mathbb{N}$ ,  $\delta \in X_2^+$ , and for all  $v_1, v_2, \dots, v_t \in \{\varepsilon, c\}$*

$$\tau(\delta) \neq \dots \text{a} v_1 \text{ab}^{3j_1-1} \text{a} \text{ab}^{3j_1} \text{a} \text{ab}^{3j_2-2} \text{a} v_2 [\dots] v_t \text{ab}^{3j_t-1} \text{a} \text{a} \dots$$

The correctness of Claim 3 follows from a straightforward combinatorial examination of the definition of  $X_2$ . Thus, and from Claim 2, it is  $\bar{\gamma}_k \notin X_2^+$  and therefore  $\bar{\gamma}_k$  must consist of variables in  $X_1$  and of variables in  $X_2$ :

*Claim 4.* For every  $k$ ,  $1 \leq k \leq \bar{m}$ ,  $|\bar{\gamma}_k| \geq 2$ .

Hence, condition (i) of Theorem 7 is satisfied for the above decomposition.

With regard to condition (iii), however, the decomposition possibly requires some modifications. We wish to have a decomposition where there is *exactly* one occurrence of an  $x_j \in X_1$  in each  $\bar{\gamma}_k$  since this variable is meant to serve as the variable  $y_k$  referred to in condition (iii) of Theorem 7. For the given decomposition, however, we can only conclude that there is *at least* one occurrence of an  $x_j \in X_1$  in each  $\bar{\gamma}_k$ . Therefore we transform it into a specific decomposition where every  $\bar{\gamma}_k$  contains exactly one  $x_j \in X_1$ . To this end, we apply two different types of operations, namely a *splitting* of certain  $\bar{\gamma}_k$  and a *redefinition* of  $X_1$  and  $X_3$ .

We first split every  $\bar{\gamma}_k$  that contains more than one occurrence of a variable from  $X_1$ , and we do so by identifying all so-called *splitting points* in  $\bar{\gamma}_k$ . Intuitively, these splitting points lead to a maximum  $s \in \mathbb{N}$  for which there exists a decomposition  $\bar{\gamma}_k = \bar{\gamma}_{k,1} \bar{\gamma}_{k,2} [\dots] \bar{\gamma}_{k,s}$  such that, for every  $k'$ ,  $1 \leq k' \leq s$ ,  $\tau(\bar{\gamma}_{k,k'})$  corresponds to  $\sigma_{\text{tf-tt},i}(\bar{\gamma}_{k,k'})$  “as far as possible”. Formally, a splitting point is an inner substring  $\delta$  of  $\bar{\gamma}_k$ , i.e.  $\bar{\gamma}_k = \bar{\gamma}_{k,l} \delta \bar{\gamma}_{k,r}$  with  $\bar{\gamma}_{k,l} = x_{j_1} x_{j_2} [\dots] x_{j_p}$  and  $\bar{\gamma}_{k,r} = x_{j_{p+1}} x_{j_{p+2}} [\dots] x_{j_{p+q}}$ ,  $p, q \in \mathbb{N}$ ,  $x_{j_1}, x_{j_2}, \dots, x_{j_{p+q}} \in X_1 \cup X_2$ , that satisfies one of the following conditions:

(1)  $\delta = \varepsilon$  and

$$\begin{aligned} \tau(\bar{\gamma}_{k,l}) &= \dots a v_1 \text{ab}^{3j_1-1} a \text{ab}^{3j_1} a [\dots] \text{ab}^{3j_p-2} a v_p \text{ab}^{3j_p-1} a a \dots \text{ and} \\ \tau(\bar{\gamma}_{k,r}) &= \dots a v_{p+1} \text{ab}^{3j_{p+1}-1} a \text{ab}^{3j_{p+1}} a [\dots] \text{ab}^{3j_{p+q}-2} a v_{p+q} \text{ab}^{3j_{p+q}-1} a a \dots \end{aligned}$$

or

(2)  $\delta = x_{j'}$ ,  $j' \in \mathbb{N}$ , and

$$\begin{aligned} \tau(\bar{\gamma}_{k,l}) &= \dots a v_1 \text{ab}^{3j_1-1} a \text{ab}^{3j_1} a [\dots] v_p \text{ab}^{3j_p-1} a \text{ab}^{3j_p} a \text{ab}^{3j'-2} a \dots \text{ and} \\ \tau(\bar{\gamma}_{k,r}) &= \dots \text{ab}^{3j'} a \text{ab}^{3j_{p+1}-2} a v_{p+1} \text{ab}^{3j_{p+1}-1} a [\dots] \text{ab}^{3j_{p+q}-2} a v_{p+q} \text{ab}^{3j_{p+q}-1} a a \dots \end{aligned}$$

for appropriate  $v_1, v_2, \dots, v_{p+q} \in \{\varepsilon, c\}$ .

For a better understanding of the definition of a splitting point, recall Claims 2 and 3. Furthermore, these claims are sufficient for verifying the following facts:

- A subpattern  $\bar{\gamma}_k$  with only one occurrence of a variable from  $X_1$  does not contain any splitting point.
- For every splitting point  $\delta$  of type 2, i.e.  $\delta = x_{j'} \in X$ , necessarily  $x_{j'} \in X_2$ .
- For two splitting points  $\delta, \delta'$ , necessarily  $\bar{\gamma}_k \neq \dots \delta \delta' \dots$ .

After *all* of the splitting points have been identified in  $\bar{\gamma}_k$ , for each of them the following *splitting operation* is performed:

- (1) If  $|\delta| = 0$  then  $\delta$  is renamed to  $\hat{\beta}$ .
- (2) If  $|\delta| = 1$  then a  $\hat{\beta} = \varepsilon$  is inserted to the right of  $\delta$ , i.e.  $\bar{\gamma}_k := \bar{\gamma}_{k,l} \delta \hat{\beta} \bar{\gamma}_{k,r}$ .

Note that, in case 2, we can arbitrarily choose to insert  $\hat{\beta}$  to the left or to the right of  $\delta$ , but it is essential to do this for all splitting points in the same way. This will be relevant for our argumentation on the crucial Claim 6.

When this has been accomplished for all splitting points then we regard the following decomposition of  $\alpha$ :

$$\alpha = \hat{\beta}_0 \hat{\gamma}_1 \hat{\beta}_1 \hat{\gamma}_2 \hat{\beta}_2 [\dots] \hat{\beta}_{\hat{m}-1} \hat{\gamma}_{\hat{m}} \hat{\beta}_{\hat{m}}$$

with  $\hat{m} \geq 1$  and

- $\hat{\beta}_k \in X_3^*$ ,  $0 \leq k \leq \hat{m}$ , where, for every  $1 \leq k' \leq \hat{m} - 1$ ,  $\hat{\beta}_{k'} = \varepsilon$  if and only if at exactly the position of  $\hat{\beta}_{k'}$  a  $\hat{\beta}$  has been inserted by a splitting operation, i.e. in this case  $\hat{\beta}_{k'}$  simply is a renaming of a  $\hat{\beta}$ , and
- $\hat{\gamma}_k \in (X_1 \cup X_2)^+$ ,  $1 \leq k \leq \hat{m}$ .

Consequently, if in some  $\bar{\gamma}_k$  there is, e.g., exactly one splitting point, i.e.  $\bar{\gamma}_k = \bar{\gamma}_{k,l} \delta \bar{\gamma}_{k,r}$ , then, for some  $h < \hat{m}$ , the splitting operation leads to  $\hat{\gamma}_h = \bar{\gamma}_{k,l}$  and  $\hat{\gamma}_{h+1} = \bar{\gamma}_{k,r}$  (in case of  $|\delta| = 0$ ) or  $\hat{\gamma}_h = \bar{\gamma}_{k,l} \delta$  and  $\hat{\gamma}_{h+1} = \bar{\gamma}_{k,r}$  (in case of  $|\delta| = 1$ ). Additionally note that  $\hat{m} \geq 1$  again follows from  $X_2 \neq \emptyset$ .

After the splitting operations we can record:

*Claim 5.* For every  $k$ ,  $1 \leq k \leq \hat{m}$ ,  $\hat{\gamma}_k$  contains exactly one occurrence of an  $x_j \in X_1$ .

Claim 5 follows from Claim 2, Claim 3 and the definition of the splitting points.

Moreover, the resulting decomposition has a second crucial property:

*Claim 6.* For every  $k, k'$ ,  $1 \leq k, k' \leq \hat{m}$ , if  $\text{var}(\hat{\gamma}_k) \cap \text{var}(\hat{\gamma}_{k'}) \cap X_1 \neq \emptyset$  then  $\hat{\gamma}_k = \hat{\gamma}_{k'}$ .

*Proof* (Claim 6). If  $|\hat{\gamma}_k| = |\hat{\gamma}_{k'}| = 1$  then Claim 6 trivially holds true. Therefore we restrict our argumentation to the case  $|\hat{\gamma}_k| \geq 2$  or  $|\hat{\gamma}_{k'}| \geq 2$ . Now assume to the contrary that there are  $k, k'$ ,  $1 \leq k, k' \leq \hat{m}$ , with  $\text{var}(\hat{\gamma}_k) \cap \text{var}(\hat{\gamma}_{k'}) \cap X_1 \neq \emptyset$  and  $\hat{\gamma}_k \neq \hat{\gamma}_{k'}$ . Because of Claim 5 we can write  $\hat{\gamma}_k$  as  $\hat{\gamma}_k = x_{j_1} x_{j_2} [\dots] x_{j_p} x_{j_\star} x_{j_{p+1}} x_{j_{p+2}} [\dots] x_{j_{p+q}}$  with  $p, q \in \mathbb{N}_0$ ,  $x_{j_\star} \in X_1$ ,  $x_{j_1}, x_{j_2}, \dots, x_{j_{p+q}} \in X_2$  and  $\hat{\gamma}_{k'}$  as  $\hat{\gamma}_{k'} = x_{j'_1} x_{j'_2} [\dots] x_{j'_r} x_{j_\star} x_{j'_{r+1}} x_{j'_{r+2}} [\dots] x_{j'_{r+s}}$  with  $r, s \in \mathbb{N}_0$ ,  $x_{j'_1}, x_{j'_2}, \dots, x_{j'_{r+s}} \in X_2$ . Note that our condition  $|\hat{\gamma}_k| \geq 2$  or  $|\hat{\gamma}_{k'}| \geq 2$  implies  $p + q + r + s \geq 1$ .

We now assume, first, that  $p = r$  and  $q = s$  (we shall examine the case where there is  $p \neq r$  or  $q \neq s$  later) and, second, w.l.o.g. that  $t \in \mathbb{N}$  is the largest number with  $j_t \neq j'_t$  and  $t \leq p$ . The latter assumption does not restrict our reasoning since, for the case that the only different variables in  $\hat{\gamma}_k, \hat{\gamma}_{k'}$  are to the right of  $x_{j_\star}$ , an analogous argumentation can be applied. Under these two assumptions, we now examine Claim 3, which says that, for every  $n$ ,  $0 \leq n \leq p - t$ ,  $\tau(x_{j_1} x_{j_2} [\dots] x_{j_{t+n}})$  does not correspond to  $\sigma_{\text{tf-tt},i}(x_{j_1} x_{j_2} [\dots] x_{j_{t+n}})$  (and of course  $\tau(x_{j'_1} x_{j'_2} [\dots] x_{j'_{t+n}})$  does not correspond to  $\sigma_{\text{tf-tt},i}(x_{j'_1} x_{j'_2} [\dots] x_{j'_{t+n}})$ ) as all of the variables under consideration are in  $X_2$ . More precisely, we may conclude that, again for every  $n$ ,  $0 \leq n \leq p - t$ ,  $\tau(x_{j_{t+n+1}} x_{j_{t+n+2}} [\dots] x_{j_\star})$  contains the right segment of  $\sigma_{\text{tf-tt},i}(x_{j_{t+n}})$  (and, additionally,  $\tau(x_{j'_{t+n+1}} x_{j'_{t+n+2}} [\dots] x_{j_\star})$  contains the right segment of  $\sigma_{\text{tf-tt},i}(x_{j'_{t+n}})$ ) since, otherwise, there would have been a splitting point somewhere between  $x_{j_t}$  and  $x_{j_\star}$  (and between  $x_{j'_t}$  and  $x_{j_\star}$ ) – this statement can be verified by a closer look at the definition of both types of splitting points, where the condition for the left subpattern  $\bar{\gamma}_{k,l}$  in the case of  $\tau(\bar{\gamma}_{k,l})$  containing the right segment of  $\sigma_{\text{tf-tt},i}(x_{j_{t+n}})$  would have led to the insertion of the said splitting point. Thus, with  $n := 0$ , this implies

$$\begin{aligned} \tau(x_{j_{t+1}} x_{j_{t+2}} [\dots] x_{j_\star}) &= \dots \text{ab}^{3j_t} \text{a} \text{ab}^{3j_{t+1}-2} \text{a} [\dots] \text{ab}^{3j_\star-1} \text{a} \text{a} \dots \\ &= \tau(x_{j'_{t+1}} x_{j'_{t+2}} [\dots] x_{j_\star}). \end{aligned}$$

On the other hand we have

$$\sigma_{\text{tf-tt},i}(x_{j'_t} x_{j'_{t+1}} x_{j'_{t+2}} [\dots] x_{j_\star}) = \dots \text{ab}^{3j'_t} \text{a} \text{ab}^{3j_{t+1}-2} \text{a} [\dots] \text{ab}^{3j_\star-1} \text{a} \text{a} \dots$$

However – since  $x_{j_\star} \in X_1$  and, consequently,  $\tau(x_{j_\star})$  contains the inner segment of  $\sigma_{\text{tf-tt},i}(x_{j_\star})$  – we know that  $\tau(x_{j_\star})$  and  $\sigma_{\text{tf-tt},i}(x_{j_\star})$  generate the *same occurrence* of the subword  $\text{ab}^{3j_\star+1} \text{a}$  in  $\tau(\alpha) = \sigma_{\text{tf-tt},i}(\alpha)$  (cf. our remarks introducing Lemma 23). Thus, we can conclude from Lemma 23 that our condition  $\tau(\alpha) = \sigma_{\text{tf-tt},i}(\alpha)$  implies  $\text{ab}^{3j_t+2} \text{a} = \text{ab}^{3j'_t+2} \text{a}$  and, hence,  $j_t = j'_t$ . This contradicts our assumption  $j_t \neq j'_t$ .

We proceed with the remaining case  $p \neq r$  or  $q \neq s$ . Due to the same reason as given above and, thus, w.l.o.g., we focus on  $p \neq r$ . Additionally and again w.l.o.g., we assume that  $p < r$ . If there is a  $t \in \mathbb{N}_0$ ,  $t < p$ , such that  $j_{p-t} \neq j'_{r-t}$  then we can apply exactly the same argument as above. Thus,  $x_{j_1} x_{j_2} [\dots] x_{j_p}$  must be a suffix of  $x_{j'_1} x_{j'_2} [\dots] x_{j'_r}$ . If  $\alpha = \hat{\gamma}_k \dots$  or  $\alpha = \dots x_{j_\#} \hat{\gamma}_k \dots$  with  $j_\# \neq j'_{r-p}$  then our argumentation again is equivalent to that on the case  $p = r$ . Hence,  $j_\# = j'_{r-p}$ . Since  $x_{j'_{r-p}} \in X_2$ ,  $\hat{\beta}_{k-1}$  must have been a splitting point separating  $x_{j'_{r-p}}$  and  $x_{j_1}$ , whereas there has not been any splitting point between  $x_{j'_{r-p}}$  and  $x_{j'_{r-p+1}}$ . Since  $x_{j_1} x_{j_2} [\dots] x_{j_p}$  is a suffix of  $x_{j'_1} x_{j'_2} [\dots] x_{j'_r}$  this contradicts the definition of splitting points.  $\square$  (Claim 6)

In a final step, we now remove all  $\hat{\gamma}_k$  with  $|\hat{\gamma}_k| = 1$ ; this type of  $\hat{\gamma}$  can occur, e.g., for  $\bar{\gamma}_{k'} = x_{j_1} x_{j_2} x_{j_3}$  with  $x_{j_1}, x_{j_3} \in X_1$  and  $x_{j_2} \in X_2$ . Consequently, for every  $\hat{\gamma}_k = x_j$ ,  $x_j \in X_1$ , we shift  $x_j$  to  $X_3$ , or, more precisely, we introduce  $X'_3 := X_3 \cup \{x_j \mid \exists k : \hat{\gamma}_k = x_j\}$  and  $X'_1 := X_1 \setminus \{x_j \mid \exists k : \hat{\gamma}_k = x_j\}$ . Note that because of Claim 5 and Claim 6 this *redefinition* operation does not affect any  $\hat{\gamma}_k$  with  $|\hat{\gamma}_k| \geq 2$ .

This leads to the final decomposition of  $\alpha$ :

$$\alpha = \beta_0 \gamma_1 \beta_1 \gamma_2 \beta_2 [\dots] \beta_{m-1} \gamma_m \beta_m$$

with  $m \geq 1$  and

- $\beta_k \in X_3^*$ ,  $0 \leq k \leq m$ , where, for every  $1 \leq k' \leq m-1$ ,  $\beta_{k'} = \varepsilon$  if and only if the variables to the right and to the left of  $\beta_{k'}$  have been split by a splitting operation and none of the resulting neighbouring  $\hat{\gamma}_{k''}$  has been removed by a shifting operation, and
- $\gamma_k \in (X'_1 \cup X_2)^+$ ,  $1 \leq k \leq m$ .

Again, this decomposition is unique, and  $m \geq 1$  is granted since  $X_2$  is not redefined and (according to our assumption)  $X_2 \neq \emptyset$ .

We conclude the proof of [Lemma 24](#) with the verification of the conditions in [Theorem 7](#):

*Claim 7. For every  $k$ ,  $1 \leq k \leq m$ ,  $|\gamma_k| \geq 2$ .*

Claim 7 is evident since the redefinition operation does not shorten or split any  $\hat{\gamma}_k$  with  $|\hat{\gamma}_k| \geq 2$ . Consequently, the above decomposition conforms with condition (i) of [Theorem 7](#). The next claim follows directly from the fact that  $X'_1$ ,  $X_2$  and  $X'_3$  are disjoint:

*Claim 8. For every  $k$ ,  $1 \leq k \leq m$ , and for every  $k'$ ,  $0 \leq k' \leq m$ ,  $\text{var}(\gamma_k) \cap \text{var}(\beta_{k'}) = \emptyset$ .*

Thus, condition (ii) of [Theorem 7](#) is satisfied as well. Since, according to the notes on Claim 7, the splitting operation does not modify any  $\hat{\gamma}_k$  with  $|\hat{\gamma}_k| \geq 2$  we can easily conclude from Claim 6:

*Claim 9. For every  $k$ ,  $1 \leq k \leq m$ ,  $\gamma_k$  contains exactly one  $x_j \in X'_1$  and, for every  $k'$ ,  $1 \leq k' \leq m$ , if  $\text{var}(\gamma_k) \cap \text{var}(\gamma_{k'}) \cap X'_1 \neq \emptyset$  then  $\gamma_k = \gamma_{k'}$ .*

This proves that condition (iii) of [Theorem 7](#) is satisfied.

Consequently, if there is an  $x_j \in \text{var}(\alpha)$  such that  $\tau(x_j) \neq \dots a v a b^{3j-1} a a \dots$ ,  $v \in \{\varepsilon, c\}$ , then  $\alpha$  is prolix. This proves the lemma.  $\square$

Consequently, when applied to succinct patterns, the ambiguity of every substitution  $\sigma_{\text{tf-tt},j}$ ,  $j \in \mathbb{N}$ , is restricted in a special manner which, in particular, is compatible with the assignment of the unique letter  $c$  to  $x_j$ . Thus, we can use the substitutions  $\sigma_{\text{tf-tt},i}$  for the definition of telltales:

**Lemma 25.** *Let  $\Sigma$  be an alphabet,  $\{a, b, c\} \subseteq \Sigma$ , and let  $L \in \text{ePAT}_{\text{tf},\Sigma}$ . For any succinct pattern  $\alpha \in \text{Pat}_{\text{tf}}$  with  $L = L_{\Sigma}(\alpha)$ , let  $T_{\alpha} := \{\sigma_{\text{tf-tt},j}(\alpha) \mid x_j \in \text{var}(\alpha)\}$ . Then  $T_{\alpha}$  is a telltale for  $L$  with respect to  $\text{ePAT}_{\text{tf},\Sigma}$ .*

**Proof.** According to [Lemma 24](#), for every  $j \in \mathbb{N}$  with  $x_j \in \text{var}(\alpha)$  and for every substitution  $\tau_j : \text{Pat}_{\text{tf}} \rightarrow \Sigma^*$  with  $\tau_j(\alpha) = \sigma_{\text{tf-tt},j}(\alpha)$ , it is  $|\tau_j(x_j)|_c = 1$ . Furthermore, by definition,  $|\tau_j(\alpha)|_c = |\alpha|_{x_j}$ . Consequently, due to [Theorem 10](#),  $T_{\alpha}$  is a telltale for  $L$  with respect to  $\text{ePAT}_{\text{tf},\Sigma}$ .  $\square$

Thus, [Lemma 25](#) shows that each terminal-free E-pattern language over an alphabet  $|\Sigma|$ ,  $|\Sigma| \geq 3$ , contains a telltale with respect to  $\text{ePAT}_{\text{tf},\Sigma}$ . As the recursive enumerability of these telltales is evident, we therefore can conclude the correctness of [Theorem 17](#) from [Theorem 3](#).

#### 4.2. Some remarks on the learnability of terminal-free E-pattern languages

Evidently, and as stated above, our reasoning in [Section 4.1](#) proves the learnability of  $\text{ePAT}_{\text{tf},\Sigma}$ ,  $|\Sigma| \geq 3$ , by a purely combinatorial argument which, in turn, is equivalent to a structural property of  $\text{ePAT}_{\text{tf},\Sigma}$ . Hence, we do not give a particular learning strategy for  $\text{ePAT}_{\text{tf},\Sigma}$ , and therefore we can merely refer to the general procedure for learnable indexed families that is provided by Angluin [2].

We regard it as a worthwhile (albeit challenging) problem for the future research on the learnability of terminal-free E-pattern languages to find a tailor-made learning strategy which more accurately matches with the characteristic of the subject. In this regard, an inconsistent learning strategy such as the procedure provided by Lange, Wiehagen [16] on the full class of NE-pattern languages, which contrary to Angluin's approach does not use any test for the membership problem, might be the overall goal of the corresponding considerations. Since the telltales identified by our reasoning



in the previous section contain rather long words (whereas those for general NE-pattern languages used in [16] simply consist of the shortest words in the respective language), it even seems inevitable to avoid membership tests as far as possible.

In addition to this, one might wish to seek for shorter telltale words in terminal-free E-pattern languages, and Example 13 gives a first idea about how such telltales might look like for selected languages. We expect that Theorem 10 can serve as a powerful tool for such a task. In general, however, we consider our telltales to be optimally chosen. Referring to the current state of knowledge on the ambiguity of morphisms in free monoids (cf. Freydenberger et al. [7]), we conjecture that the assignment of a number of distinct segments  $ab^m a$ ,  $m \in \mathbb{N}$ , to each variable is the only uniform method for generating those “moderately ambiguous” words required by Theorem 10. If we now replace  $\sigma_{\text{tf-tt},i}$  by a substitution mapping the variables on just two distinct segments then this shorter substitution fails in generating telltales. In order to verify this statement we regard the following substitution which omits the inner segment that, for every  $x_j \in X$ , is contained in  $\sigma_{\text{tf-tt},i}(x_j)$ :

**Definition 26.** Let  $\Sigma$  be an alphabet,  $\{a, b, c\} \subseteq \Sigma$ . Then, for every  $i, j \in \mathbb{N}$ , the substitution  $\sigma_{2\text{-seg},i} : \text{Pat}_{\text{tf}} \longrightarrow \Sigma^*$  is given by

$$\sigma_{2\text{-seg},i}(x_j) := \begin{cases} ab^{2j-1} a c ab^{2j} a, & i = j, \\ ab^{2j-1} a ab^{2j} a, & \text{else.} \end{cases}$$

As explained above, the following lemma states that, for every alphabet  $\Sigma$  which consists of at least three distinct letters, there exists a terminal-free E-pattern language  $L \subseteq \Sigma^*$  such that the substitutions  $\sigma_{2\text{-seg},i}$  are not appropriate for defining a telltale for  $L$  with respect to  $\text{ePAT}_{\text{tf},\Sigma}$ :

**Lemma 27.** Let  $\Sigma$  be an alphabet,  $\{a, b, c\} \subseteq \Sigma$ . Then there exists a succinct pattern  $\alpha \in \text{Pat}_{\text{tf}}$  such that the set  $W_\alpha := \{\sigma_{2\text{-seg},j}(\alpha) \mid x_j \in \text{var}(\alpha)\}$  is not a telltale for  $L_\Sigma(\alpha)$  with respect to  $\text{ePAT}_{\text{tf},\Sigma}$ .

**Proof.** Let the pattern  $\alpha \in \text{Pat}_{\text{tf}}$  be given by

$$\alpha := x_1 x_2 x_3 x_4 x_1 x_2 x_3 x_4 x_5 x_2 x_6 x_5 x_7 x_8 x_6 x_8 x_6 x_9 x_7 x_9 x_7 x_{10} x_4 x_{10} x_{11} x_{12} x_4 x_{12} x_{11}.$$

The succinctness of  $\alpha$  can be straightforward verified by Theorem 7. In addition to this, let the substitution  $\tau_3$  be given by

$$\begin{aligned} \tau_3(x_1) &:= aba ab^2 a ab^3 a ab^4 a ab^5 a c ab^3, \\ \tau_3(x_2) &:= b^3 a ab^3, \\ \tau_3(x_3) &:= \varepsilon, \\ \tau_3(x_4) &:= b^4 a ab^8 a, \\ \tau_3(x_5) &:= ab^9 a ab^{10} a a, \\ \tau_3(x_6) &:= ba ab^{11} a ab^{12} a, \\ \tau_3(x_7) &:= b^{13} a ab^{14} a, \\ \tau_3(x_8) &:= ab^{15} a ab^{15}, \\ \tau_3(x_9) &:= ab^{17} a ab^{18} a a, \\ \tau_3(x_{10}) &:= ab^{19} a ab^{20} a ab^3, \\ \tau_3(x_{11}) &:= b^{18} a ab^{22} a, \\ \tau_3(x_{12}) &:= ab^{23} a ab^{24} a ab^3. \end{aligned}$$

Then  $\tau_3$  and  $\sigma_{2\text{-seg},3}$  generate the same word when applied to  $\alpha$ :

$$\begin{aligned}
\sigma_{2\text{-seg},3}(\alpha) &= \underbrace{\overbrace{aba}^{\sigma_{2\text{-seg},3}(x_1)} \overbrace{ab^2a}^{\sigma_{2\text{-seg},3}(x_2)} \overbrace{ab^3a}^{\sigma_{2\text{-seg},3}(x_3)} \overbrace{ab^4a}^{\sigma_{2\text{-seg},3}(x_4)} \overbrace{ab^5a}^{\sigma_{2\text{-seg},3}(x_5)} c \overbrace{ab^3}^{\sigma_{2\text{-seg},3}(x_6)}}_{\tau_3(x_1)} \underbrace{\overbrace{b^3a}^{\sigma_{2\text{-seg},3}(x_7)} \overbrace{ab^3}^{\sigma_{2\text{-seg},3}(x_8)}}_{\tau_3(x_2)} \underbrace{\overbrace{b^4a}^{\sigma_{2\text{-seg},3}(x_9)} \overbrace{ab^8a}^{\sigma_{2\text{-seg},3}(x_{10})}}_{\tau_3(x_4)} \\
&\quad \underbrace{\overbrace{aba}^{\sigma_{2\text{-seg},3}(x_1)} \overbrace{ab^2a}^{\sigma_{2\text{-seg},3}(x_2)} \overbrace{ab^3a}^{\sigma_{2\text{-seg},3}(x_3)} \overbrace{ab^4a}^{\sigma_{2\text{-seg},3}(x_4)} \overbrace{ab^5a}^{\sigma_{2\text{-seg},3}(x_5)} c \overbrace{ab^3}^{\sigma_{2\text{-seg},3}(x_6)}}_{\tau_3(x_1)} \underbrace{\overbrace{b^3a}^{\sigma_{2\text{-seg},3}(x_7)} \overbrace{ab^3}^{\sigma_{2\text{-seg},3}(x_8)}}_{\tau_3(x_2)} \underbrace{\overbrace{b^4a}^{\sigma_{2\text{-seg},3}(x_9)} \overbrace{ab^8a}^{\sigma_{2\text{-seg},3}(x_{10})}}_{\tau_3(x_4)} \\
&\quad \underbrace{\overbrace{ab^9a}^{\sigma_{2\text{-seg},3}(x_5)} \overbrace{ab^{10}a}^{\sigma_{2\text{-seg},3}(x_6)} a}_{\tau_3(x_5)} \underbrace{\overbrace{b^3a}^{\sigma_{2\text{-seg},3}(x_7)} \overbrace{ab^3}^{\sigma_{2\text{-seg},3}(x_8)}}_{\tau_3(x_2)} \underbrace{\overbrace{ba}^{\sigma_{2\text{-seg},3}(x_9)} \overbrace{ab^{11}a}^{\sigma_{2\text{-seg},3}(x_{10})} \overbrace{ab^{12}a}^{\sigma_{2\text{-seg},3}(x_{11})}}_{\tau_3(x_6)} \underbrace{\overbrace{ab^9a}^{\sigma_{2\text{-seg},3}(x_5)} \overbrace{ab^{10}a}^{\sigma_{2\text{-seg},3}(x_6)} a}_{\tau_3(x_5)} \underbrace{\overbrace{b^{13}a}^{\sigma_{2\text{-seg},3}(x_7)} \overbrace{ab^{14}a}^{\sigma_{2\text{-seg},3}(x_8)}}_{\tau_3(x_7)} \\
&\quad \underbrace{\overbrace{ab^{15}a}^{\sigma_{2\text{-seg},3}(x_8)} \overbrace{ab^{15}a}^{\sigma_{2\text{-seg},3}(x_9)}}_{\tau_3(x_8)} \underbrace{\overbrace{ba}^{\sigma_{2\text{-seg},3}(x_{10})} \overbrace{ab^{11}a}^{\sigma_{2\text{-seg},3}(x_{11})} \overbrace{ab^{12}a}^{\sigma_{2\text{-seg},3}(x_{12})}}_{\tau_3(x_6)} \underbrace{\overbrace{ab^{15}a}^{\sigma_{2\text{-seg},3}(x_8)} \overbrace{ab^{15}a}^{\sigma_{2\text{-seg},3}(x_9)}}_{\tau_3(x_8)} \underbrace{\overbrace{ba}^{\sigma_{2\text{-seg},3}(x_{10})} \overbrace{ab^{11}a}^{\sigma_{2\text{-seg},3}(x_{11})} \overbrace{ab^{12}a}^{\sigma_{2\text{-seg},3}(x_{12})}}_{\tau_3(x_6)} \\
&\quad \underbrace{\overbrace{ab^{17}a}^{\sigma_{2\text{-seg},3}(x_9)} \overbrace{ab^{18}a}^{\sigma_{2\text{-seg},3}(x_{10})} a}_{\tau_3(x_9)} \underbrace{\overbrace{b^{13}a}^{\sigma_{2\text{-seg},3}(x_7)} \overbrace{ab^{14}a}^{\sigma_{2\text{-seg},3}(x_8)}}_{\tau_3(x_7)} \underbrace{\overbrace{ab^{17}a}^{\sigma_{2\text{-seg},3}(x_9)} \overbrace{ab^{18}a}^{\sigma_{2\text{-seg},3}(x_{10})} a}_{\tau_3(x_9)} \underbrace{\overbrace{b^{13}a}^{\sigma_{2\text{-seg},3}(x_7)} \overbrace{ab^{14}a}^{\sigma_{2\text{-seg},3}(x_8)}}_{\tau_3(x_7)} \\
&\quad \underbrace{\overbrace{ab^{19}a}^{\sigma_{2\text{-seg},3}(x_{10})} \overbrace{ab^{20}a}^{\sigma_{2\text{-seg},3}(x_{11})} a}_{\tau_3(x_{10})} \underbrace{\overbrace{b^4a}^{\sigma_{2\text{-seg},3}(x_4)} \overbrace{ab^8a}^{\sigma_{2\text{-seg},3}(x_5)}}_{\tau_3(x_4)} \underbrace{\overbrace{ab^{19}a}^{\sigma_{2\text{-seg},3}(x_{10})} \overbrace{ab^{20}a}^{\sigma_{2\text{-seg},3}(x_{11})} a}_{\tau_3(x_{10})} \underbrace{\overbrace{b^{18}a}^{\sigma_{2\text{-seg},3}(x_{11})} \overbrace{ab^{22}a}^{\sigma_{2\text{-seg},3}(x_{12})}}_{\tau_3(x_{11})} \\
&\quad \underbrace{\overbrace{ab^{23}a}^{\sigma_{2\text{-seg},3}(x_{12})} \overbrace{ab^{24}a}^{\sigma_{2\text{-seg},3}(x_{13})} a}_{\tau_3(x_{12})} \underbrace{\overbrace{b^4a}^{\sigma_{2\text{-seg},3}(x_4)} \overbrace{ab^8a}^{\sigma_{2\text{-seg},3}(x_5)}}_{\tau_3(x_4)} \underbrace{\overbrace{ab^{23}a}^{\sigma_{2\text{-seg},3}(x_{12})} \overbrace{ab^{24}a}^{\sigma_{2\text{-seg},3}(x_{13})} a}_{\tau_3(x_{12})} \underbrace{\overbrace{b^{18}a}^{\sigma_{2\text{-seg},3}(x_{11})} \overbrace{ab^{22}a}^{\sigma_{2\text{-seg},3}(x_{12})}}_{\tau_3(x_{11})} \\
&= \tau_3(\alpha).
\end{aligned}$$

Moreover,  $\tau_3(x_3) = \varepsilon$  and, for every  $j \in \mathbb{N}$  with  $x_j \in \text{var}(\alpha) \setminus \{x_3\}$  and for every letter  $A$  occurring in  $\sigma_{2\text{-seg},j}(x_3)$ ,  $|\sigma_{2\text{-seg},j}(x_3)|_A \geq 2$ . Thus, with regard to  $x_3$ ,  $W_\alpha$  does not satisfy the characteristic criterion given in [Theorem 10](#). This proves the lemma.  $\square$

Recall that [Example 14](#) gives the analogous result for a substitution which maps each variable onto a word that just consists of a single segment.

Thus, [Example 14](#), [Lemmas 25](#) and [27](#) suggest that, in general, the rather long and very special words generated by the  $\sigma_{\text{tf-tt},i}$  are required for drawing unequivocal conclusions on the respective generating pattern under consideration. If there is no option to switch to shorter significant words then we expect this to cause major problems for *stochastic finite learning* of  $\text{ePAT}_{\text{tf}}$  (as introduced by Rossmanith, Zeugmann [\[33\]](#) with respect to the full class of NE-pattern languages) – even if it is possible to give a learning strategy for the terminal-free E-pattern languages over suitable alphabets that is not based on exhaustive membership tests.

## 5. Inductive inference of general E-pattern languages over small alphabets

In the present section, we shall examine whether the positive learnability result on  $\text{ePAT}_{\text{tf}}$  presented in [Theorem 17](#) can be extended to general E-pattern languages.

At first glance – due to the previous insights into *regular E-pattern languages* (generated by patterns where every variable occurs at most once) gained by Shinohara [\[36\]](#) – such an extension seems to be possible. Since, according to the cited work, the class of regular E-pattern languages is inferrable from positive data, one might interpret this result in such a way that, in the limit, the shape and position of the terminal substrings in a generating pattern can be inferred from its language. Thus, roughly speaking, it seems to be an auspicious strategy to present the text for a general E-pattern language to a (possibly modified) learner for regular E-pattern languages, then use its output to identify those parts of the words that have been generated by the substitution of variables and, finally, have these subwords read by a learner for terminal-free E-pattern languages, so as to specify the dependencies of the variables in the generating

pattern. The main result of the present section, however, states that such an approach necessarily fails since the full class of E-pattern languages is not learnable (at least in the case that the corresponding alphabet consists of three or four distinct letters):

**Theorem 28.** *Let  $\Sigma$  be an alphabet  $|\Sigma| \in \{3, 4\}$ . Then  $\text{ePAT}_\Sigma$  is not inferable from positive data.*

The proof for Theorem 28 is given in Section 5.1.

Contrary to the reasons for the discontinuous properties of terminal-free E-pattern languages with regard to their learnability discussed in Section 4, we feel those for this second discontinuity (i.e. the non-extensibility of Theorem 17) identified in the present paper to be well understood. In the subsequent section we demonstrate that terminal-preserving morphisms lead to a more involved ambiguity of words than standard morphisms, and therefore we can give E-pattern languages which do not have a telltale with respect  $\text{ePAT}_\Sigma$  for the alphabets  $\Sigma$  under consideration.

We now summarise the current state of knowledge on the learnability of the full class of E-pattern languages:

**Corollary 29.** *Let  $\Sigma$  be an alphabet. Then  $\text{ePAT}_\Sigma$  is inferable from positive data if  $|\Sigma| \in \{1, \infty\}$ , and it is not inferable from positive data if  $|\Sigma| \in \{2, 3, 4\}$ .*

**Proof.** With regard to unary and infinite alphabets, Corollary 29 is proven by Mitchell [24], and the negative results on the other alphabet sizes are given in Theorems 16 and 28.  $\square$

Resulting from the considerations in the subsequent section, we conjecture that there is no discontinuity in the learnability of E-pattern languages over alphabets with at least five distinct letters:

**Conjecture 30.** *Let  $\Sigma_1, \Sigma_2$  be arbitrary finite alphabets with  $|\Sigma_1| \geq 5$  and  $|\Sigma_2| \geq 5$ . Then  $\text{ePAT}_{\Sigma_1}$  is inferable from positive data if and only if  $\text{ePAT}_{\Sigma_2}$  is inferable from positive data.*

We expect that any progress on the open cases requires deep insights into the ambiguity of terminal-preserving morphisms over alphabets with five or more letters. As shown by Reidenbach [28], this topic is closely related to the equivalence problem for E-pattern languages.

### 5.1. Proof of Theorem 28

As to be shown in the present section, the non-learnability of  $\text{ePAT}_{\text{tf}, \Sigma}$ ,  $|\Sigma| \in \{3, 4\}$ , results from the non-learnability of a natural subclass. This class is generated by the set of all patterns which do not contain at least two distinct terminal symbols occurring in the alphabet:

**Definition 31.** Let  $\Sigma$  be an alphabet,  $|\Sigma| \geq 2$ . Then a pattern  $\alpha \in \text{Pat}_\Sigma$  is said to be *quasi-terminal-free* (on  $\Sigma$ ) provided that  $|\Sigma| - |\text{term}(\alpha)| \geq 2$ . The set of all quasi-terminal-free patterns on  $\Sigma$  is denoted by  $\text{Pat}_{\text{q-tf}, \Sigma}$ . An E-pattern language  $L$  is called *quasi-terminal-free* (on  $\Sigma$ ) if there exists a pattern  $\alpha \in \text{Pat}_{\text{q-tf}, \Sigma}$  with  $L = L_\Sigma(\alpha)$ . The class of all quasi-terminal-free E-pattern languages on  $\Sigma$  is referred to by  $\text{ePAT}_{\text{q-tf}, \Sigma}$ .

It can be easily derived from Theorem 3 that any considerations on the inferability of a class of languages essentially depend on insights into the inclusion problem for this class. Therefore, our subsequent reasoning on Theorem 28 greatly benefits from the fact that the inclusion of two quasi-terminal-free E-pattern languages is a well-understood topic provided that these languages are generated by *similar* patterns:

**Theorem 32** (Ohlebusch, Ukkonen [25]). *Let  $\Sigma$  be an alphabet,  $|\Sigma| \geq 2$ , and let  $\alpha, \beta \in \text{Pat}_{\text{q-tf}, \Sigma}$  be similar patterns. Then  $L_\Sigma(\beta) \subseteq L_\Sigma(\alpha)$  if and only if there exists a terminal-preserving morphism  $\phi : (\Sigma \cup X)^* \rightarrow (\Sigma \cup X)^*$  with  $\phi(\alpha) = \beta$ .*

Note that Theorem 32 explains the term “quasi-terminal-free” since it can be seen as the natural extension of Theorem 2, which describes the inclusion of terminal-free E-pattern languages and which is crucial for our analysis of the learnability of  $\text{ePAT}_{\text{tf}}$  (see Section 4.1 and, additionally, the proof for Theorem 10).

We now begin our proof for Theorem 28 by the definition of the corresponding quasi-terminal-free example patterns:

**Definition 33.** The patterns  $\alpha_{abc}$  and  $\alpha_{abcd}$  are given by

$$\begin{aligned}\alpha_{abc} &:= x_1 a x_2 x_3^2 x_4^2 x_5^2 x_6^2 a x_7 a x_2 x_8^2 x_4^2 x_5^2 x_6^2, \\ \alpha_{abcd} &:= x_1 a x_2 x_3^2 x_4^2 x_5^2 x_6^2 x_7^2 x_8 b x_9 a x_2 x_{10}^2 x_4^2 x_5^2 x_6^2 x_{11}^2 x_8 b x_{12}.\end{aligned}$$

The pattern  $\alpha_{abc}$  is used in Lemma 34 for the proof of Theorem 28 in the case of  $|\Sigma_1| = 3$ , and  $\alpha_{abcd}$  is examined in Lemma 35 with regard to  $|\Sigma_2| = 4$ . In these lemmata we show that  $L_{\Sigma_1}(\alpha_{abc})$  does not have a telltale with respect to  $\text{ePAT}_{q\text{-tf}, \Sigma_1}$  and that  $L_{\Sigma_2}(\alpha_{abcd})$  does not have a telltale with respect to  $\text{ePAT}_{q\text{-tf}, \Sigma_2}$ .

First, due to the intricacy of these patterns, we consider it helpful for the understanding of the proofs of the lemmata to briefly discuss the meaning of some of their variables and terminal symbols in our reasoning; we focus on  $\alpha_{abc}$  since  $\alpha_{abcd}$  is a natural extension thereof. In a first step, our argumentation on the lemmata utilises the insight that, with regard to *terminal-free* E-pattern languages, the ambiguity of a word decides on the question of whether this word can be a useful part of a telltale (cf. Theorem 10). More precisely, concerning the terminal-free pattern  $\alpha_0 := x_4^2 x_5^2 x_6^2$  it is explained by Theorem 10 and, in particular, by Example 13 that any telltale for  $L(\alpha_0)$  necessarily has to contain words which consist of *three* distinct letters since these words, first, must be “reasonably unambiguous” with respect to  $\alpha_0$  (which can only be guaranteed by words over at least two different letters) and, second, have to contain an additional unique letter that is unequivocally related to a distinct variable in  $\alpha_0$ . Hence – and because of the fact that  $\alpha_0$  is a subpattern of  $\alpha_{abc}$  – there must be a word in any telltale for  $L_{\Sigma_1}(\alpha_{abc})$  that is generated by a substitution  $\sigma$  which maps  $\alpha_0$  onto a word over all letters provided by  $\Sigma_1 := \{a, b, c\}$ ; evidently, this means that  $\sigma(\alpha_0)$  necessarily contains the letter *a*. Therefore, in a second step, we can rely on the fact that, for the prefix  $\alpha_1 := x_1 a x_2 x_3^2 \alpha_0$  of  $\alpha_{abc}$ , now  $\sigma(\alpha_1)$  is ambiguous with respect to  $\alpha_1$  and may always be generated by a second substitution  $\tau$  with  $\tau(\alpha_0) := \varepsilon$ ,  $\tau(x_1) := \sigma(x_1 a x_2 x_3^2)[\sigma(\alpha_0)/a]$ ,  $\tau(x_2) := [a \setminus \sigma(\alpha_0)]$ . Due to the existence of  $\tau$ , in turn, we can give an inverse substitution leading to a tailor-made pattern that assuredly can be part of a passe-partout. Thus, for  $\alpha_1$  we can state a gap between, on the one hand, the need of substituting  $\alpha_0$  by three different letters and, on the other hand, the ambiguity of all words that conform to this requirement. However, due to the unique variable  $x_2$  in  $\alpha_1$ , this pattern is prolix, and the language generated by  $\alpha_1$  equals that of  $\alpha_2 := x_1 a x_2$  (cf. Theorem 32), turning the core subpattern  $\alpha_0$  of  $\alpha_1$  to be redundant. Since our formal argumentation on the subsequent Lemma 34 requires a *succinct* pattern which, nevertheless, allows for the ambiguity of words described above, the variable  $x_2$  and the subpattern  $\alpha_0$  have to occur at least twice in the pattern. This is guaranteed by introducing the suffix  $\alpha'_1 := x_7 a x_2 x_8^2 \alpha_0$ , so that our first crucial example pattern finally reads  $\alpha_{abc} = \alpha_1 a \alpha'_1$ .

With regard to  $\alpha_{abcd}$ , the underlying principle is similar. As stated above, three distinct letters are needed for an appropriate “telltale substitution”  $\sigma$  of  $\alpha_0$ . However, if *b, c, d* are chosen as these letters, the abovementioned ambiguity of  $\sigma(\alpha_1)$ , which depends on an occurrence of the letter *a* in  $\sigma(\alpha)$ , cannot be guaranteed. Hence, in  $\alpha_{abcd}$ , the subpattern  $\alpha_1$  is extended to  $\hat{\alpha}_1 := \alpha_1 x_7^2 x_8 b x_9$ , such that every  $\sigma(\hat{\alpha}_1)$  is ambiguous as soon as  $\sigma(\alpha_0)$  contains the letters *a* or *b*. Furthermore, due to the reasons described above, a modification of  $\hat{\alpha}_1$  serves as suffix of  $\alpha_{abcd}$ , namely  $\hat{\alpha}'_1 := x_9 a x_2 x_{10}^2 \alpha_0 x_{11}^2 x_8 b x_{12}$ . Contrary to the structure of  $\alpha_{abc}$ , the prefix  $\hat{\alpha}_1$  and the suffix  $\hat{\alpha}'_1$  in this case are not separated by a terminal symbol, but they are overlapping.

With regard to  $|\Sigma| = 3$ , we now specify and formalise the approach discussed above:

**Lemma 34.** Let  $\Sigma := \{a, b, c\}$ . Then for  $\alpha_{abc}$  and for every finite  $W \subset L_{\Sigma}(\alpha_{abc})$  there exists a passe-partout  $\beta \in \text{Pat}_{q\text{-tf}, \Sigma}$ .

**Proof.** If  $W$  is empty then Lemma 34 holds trivially. Hence, let  $W := \{w_1, w_2, \dots, w_n\}$  be non-empty. Then, as  $W \subset L_{\Sigma}(\alpha_{abc})$ , for every  $w_i \in W$  there exists a substitution  $\sigma_i$  satisfying  $\sigma_i(\alpha_{abc}) = w_i$ . Using these  $\sigma_i$  the following procedure constructs a passe-partout  $\beta \in \text{Pat}_{q\text{-tf}, \Sigma}$ :

Initially, we define

$$\beta_0 := \gamma_{1,0} a \gamma_{2,0} \gamma_{3,0}^2 \gamma_{4,0}^2 \gamma_{5,0}^2 \gamma_{6,0}^2 a \gamma_{7,0} a \gamma_{2,0} \gamma_{8,0}^2 \gamma_{4,0}^2 \gamma_{5,0}^2 \gamma_{6,0}^2$$

with  $\gamma_{j,0} := \varepsilon$  for every  $j$ ,  $1 \leq j \leq 8$ .

For every  $w_i \in W$  we define an inverse substitution  $\bar{\sigma}_i : \Sigma^* \longrightarrow X^*$  by

$$\bar{\sigma}_i(A) := \begin{cases} x_{3i-2}, & A = a, \\ x_{3i-1}, & A = b, \\ x_{3i}, & A = c. \end{cases}$$

For every  $i = 1, 2, 3, \dots, n$  we now consider the following cases:

Case 1: There is no  $A \in \Sigma$  with  $|\sigma_i(x_6)|_A = 1$  and  $|\sigma_i(\text{tf}(\alpha_{abc}))|_A = 4$ .

Define  $\gamma_{j,i} := \gamma_{j,i-1} \bar{\sigma}_i(\sigma_i(x_j))$  for every  $j$ ,  $1 \leq j \leq 8$ .

Case 2: There is an  $A \in \Sigma$  with  $|\sigma_i(x_6)|_A = 1$  and  $|\sigma_i(\text{tf}(\alpha_{abc}))|_A = 4$ .

Case 2.1:  $A = a$ .

$$\begin{aligned} \text{Define } \gamma_{1,i} &:= \gamma_{1,i-1} \bar{\sigma}_i(\sigma_i(x_1 a x_2 x_3^2 x_4^2 x_5^2)) \bar{\sigma}_i([\sigma_i(x_6^2)/a]), \\ \gamma_{2,i} &:= \gamma_{2,i-1} \bar{\sigma}_i([a \setminus \sigma_i(x_6^2)]), \\ \gamma_{7,i} &:= \gamma_{7,i-1} \bar{\sigma}_i(\sigma_i(x_7 a x_2 x_8^2 x_4^2 x_5^2)) \bar{\sigma}_i([\sigma_i(x_6^2)/a]), \\ \gamma_{j,i} &:= \gamma_{j,i-1}, \quad j \in \{3, 4, 5, 6, 8\}. \end{aligned}$$

Case 2.2:  $A = b$ , and therefore  $\sigma_i(x_4^2 x_5^2) \in \{a, c\}^*$ .

Case 2.2.1:  $\sigma_i(x_4^2 x_5^2) \in \{a\}^* \cup \{c\}^*$ .

$$\begin{aligned} \text{Define } \gamma_{4,i} &:= \gamma_{4,i-1} \bar{\sigma}_i(\sigma_i(x_4 x_5)), \\ \gamma_{5,i} &:= \gamma_{5,i-1} \bar{\sigma}_i(\sigma_i(x_6)), \\ \gamma_{6,i} &:= \gamma_{6,i-1}, \\ \gamma_{j,i} &:= \gamma_{j,i-1} \bar{\sigma}_i(\sigma_i(x_j)), \quad j \in \{1, 2, 3, 7, 8\}. \end{aligned}$$

Case 2.2.2:  $\sigma_i(x_4^2 x_5^2) \in \{a, c\}^+ \setminus (\{a\}^+ \cup \{c\}^+)$ .

$$\begin{aligned} \text{Define } \gamma_{1,i} &:= \gamma_{1,i-1} \bar{\sigma}_i(\sigma_i(x_1 a x_2 x_3^2)) \bar{\sigma}_i([\sigma_i(x_4^2 x_5^2)/a]), \\ \gamma_{2,i} &:= \gamma_{2,i-1} \bar{\sigma}_i([a \setminus \sigma_i(x_4^2 x_5^2 x_6^2)]), \\ \gamma_{7,i} &:= \gamma_{7,i-1} \bar{\sigma}_i(\sigma_i(x_7 a x_2 x_8^2)) \bar{\sigma}_i([\sigma_i(x_4^2 x_5^2)/a]), \\ \gamma_{j,i} &:= \gamma_{j,i-1}, \quad j \in \{3, 4, 5, 6, 8\}. \end{aligned}$$

Case 2.3:  $A = c$ , and therefore  $\sigma_i(x_4^2 x_5^2) \in \{a, b\}^*$ .

If, in the conditions of Cases 2.2.1 and 2.2.2,  $c$  is replaced by  $b$  then the subcases and definitions of Case 2.2 exactly corresponds to what is appropriate for Case 2.3.

Finally, define

$$\beta_i := \gamma_{1,i} a \gamma_{2,i} \gamma_{3,i}^2 \gamma_{4,i}^2 \gamma_{5,i}^2 \gamma_{6,i}^2 a \gamma_{7,i} a \gamma_{8,i}^2 \gamma_{4,i}^2 \gamma_{5,i}^2 \gamma_{6,i}^2.$$

When this has been accomplished for every  $i$ ,  $1 \leq i \leq n$ , then define  $\beta := \beta_n$ .

Now, in order to conclude the proof, the following has to be shown:  $\beta$  is a passe-partout for  $\alpha_{abc}$  and  $W$ , i.e.

- (1)  $W \subseteq L_{\Sigma}(\beta)$  and
- (2)  $L_{\Sigma}(\beta) \subset L_{\Sigma}(\alpha_{abc})$ .

ad 1. For every  $i$ ,  $1 \leq i \leq n$ , we define a substitution  $\sigma'_i$  by

$$\sigma'_i(x_j) := \begin{cases} a, & j = 3i - 2, \\ b, & j = 3i - 1, \\ c, & j = 3i, \\ \varepsilon, & \text{else.} \end{cases}$$

We now demonstrate that, for every  $i$ ,  $1 \leq i \leq n$ ,  $\sigma'_i(\beta) = w_i$ . We start by the observation that, in a sense,  $\sigma'_i$  is simply the inverse morphism to  $\bar{\sigma}_i$ . This is guaranteed by the fact that  $\sigma'_i$  maps a variable  $x_j$  onto  $\varepsilon$  whenever the occurrences of  $x_j$  in  $\beta$  result from some inverse substitution  $\bar{\sigma}_{i'}$  satisfying  $i' \neq i$ . Hence, we only need to verify that, for every  $i$ ,  $1 \leq i \leq n$ , the relevant variables in  $\beta$  occur in the same order as the corresponding letters in  $\sigma_i(\alpha_{abc})$ . If  $\sigma_i$



satisfies Case 1 then this holds immediately as, for every  $x_j \in \text{var}(\alpha_{\text{abc}})$ , we apply  $\bar{\sigma}_i$  simply to  $\sigma_i(x_j)$ . If  $\sigma_i$  satisfies Case 2 then  $w_i$  is ambiguous with respect to  $\alpha_{\text{abc}}$  (which, e. g., can be verified by a closer look at our explanations on the structure of  $\alpha_{\text{abc}}$  given below [Definition 33](#)). In this case, the application of  $\bar{\sigma}_i$  does not correspond to  $\sigma_i$  but to a particular substitution  $\tau$  satisfying  $\tau(\alpha_{\text{abc}}) = \sigma_i(\alpha_{\text{abc}})$ . Thus, for every  $i$ ,  $1 \leq i \leq n$ , the order of the variables in  $\beta$  that are generated by the application of  $\bar{\sigma}_i$  to  $w_i$  equals the order of the respective letters in  $w_i$ , and therefore  $\sigma'_i(\beta) = w_i$ , which immediately implies  $w_i \in L_\Sigma(\beta)$ . Consequently,  $W \subseteq L_\Sigma(\beta)$ .

ad 2. As there exists a terminal-preserving morphism  $\phi : (\Sigma \cup X)^* \longrightarrow (\Sigma \cup X)^*$  with  $\phi(\alpha_{\text{abc}}) = \beta$ , which is given by  $\phi(x_j) := \gamma_{j,n}$  for every  $x_j \in \text{var}(\alpha_{\text{abc}})$ ,  $L_\Sigma(\beta) \subseteq L_\Sigma(\alpha_{\text{abc}})$  follows directly from [Theorem 1](#).

We now prove that  $L_\Sigma(\beta)$  is a proper subset of  $L_\Sigma(\alpha_{\text{abc}})$ . If  $\alpha_{\text{abc}}$  and  $\beta$  are not similar then it can be easily verified that one of the following words is in  $L_\Sigma(\alpha_{\text{abc}}) \setminus L_\Sigma(\beta)$ : baaa, abbaa, aaba or aaabb. Consequently,  $L_\Sigma(\alpha_{\text{abc}}) \neq L_\Sigma(\beta)$  which implies  $L_\Sigma(\beta) \subset L_\Sigma(\alpha_{\text{abc}})$ . Hence, we turn our attention to the case that  $\alpha_{\text{abc}}$  and  $\beta$  are similar. In this case, we wish to use [Theorem 2](#) for proving that  $L_\Sigma(\alpha_{\text{abc}}) \not\subseteq L_\Sigma(\beta)$ . To this end, we have to show that there is no morphism  $\psi : (\Sigma \cup X)^* \longrightarrow (\Sigma \cup X)^*$  with  $\psi(\beta) = \alpha_{\text{abc}}$ . For that purpose, assume to the contrary that there is such a morphism  $\psi$ . Then, as there is no variable in  $\text{var}(\alpha_{\text{abc}})$  with more than four occurrences in  $\alpha_{\text{abc}}$ ,  $\psi(x_k) = \varepsilon$  for all  $x_k \in \text{var}(\beta)$  with  $|\beta|_{x_k} \geq 5$ . With regard to the variables in  $\text{var}(\gamma_{6,n})$ , this means the following: If every letter in  $\sigma_i(x_6)$  occurs more than four times in  $\sigma_i(\text{tf}(\alpha_{\text{abc}}))$  then Case 1 is satisfied and, consequently, every variable that is added to  $\gamma_{6,i}$  occurs at least five times in  $\beta$ . If any letter A in  $\sigma_i(x_6)$  occurs exactly four times in  $\sigma_i(\text{tf}(\alpha_{\text{abc}}))$  – and, obviously, it must be at least four times as  $|\alpha_{\text{abc}}|_{x_6} = 4$  – then Case 2 is applied, which, enabled by the ambiguity of  $w_i$  in that case, arranges the newly added components of  $\gamma_{6,i}$  such that  $\bar{\sigma}_i(\sigma_i(A))$  is shifted to a different  $\gamma_{j,i}$ . Consequently,  $|\beta|_{x_k} \geq 5$  for all  $x_k \in \text{var}(\gamma_{6,n})$  and, therefore,  $\psi(\gamma_{6,n}) = \varepsilon \neq x_6$ . Hence, we analyse whether or not  $\text{var}(\alpha_{\text{abc}})$  contains an anchor variable  $x_{j'}$  with respect to  $\phi$  and  $\psi$  (cf. [Lemma 4](#), which can be canonically extended to terminal-preserving morphisms). Evidently,  $j' \notin \{1, 7\}$ ; for  $j' \in \{3, 4, 5, 8\}$ ,  $x_{j'}$  being an anchor variable implies that, for some variables  $x_k, x_{k'}$ ,  $\psi(\gamma_{j',n}^2) = \dots x_k x_{k'} \dots x_k x_{k'} \dots$ , but there is no substring in  $\alpha_{\text{abc}}$  that equals the given shape of  $\psi(\gamma_{j',n}^2)$ . Finally,  $x_2$  cannot be an anchor variable since  $\psi(\gamma_{2,n})$  had to equal both  $x_2 x_3 \dots$  and  $x_2 x_8 \dots$ . Consequently, there is no anchor variable in  $\text{var}(\alpha_{\text{abc}})$ . This contradicts  $\psi(\gamma_{6,n}) = \varepsilon \neq x_6$  and therefore the assumption is incorrect. Thus,  $L_\Sigma(\beta) \not\subseteq L_\Sigma(\alpha_{\text{abc}})$  and, finally,  $L_\Sigma(\beta) \subset L_\Sigma(\alpha_{\text{abc}})$ .  $\square$

We proceed with the analogous reasoning on  $|\Sigma| = 4$ :

**Lemma 35.** *Let  $\Sigma := \{a, b, c, d\}$ . Then for  $\alpha_{\text{abcd}}$  and for every finite  $W \subset L_\Sigma(\alpha_{\text{abcd}})$  there exists a passe-partout  $\beta \in \text{Pat}_{\text{q-tf}, \Sigma}$ .*

**Proof.** We can argue similar to the proof of [Lemma 34](#): If  $W$  is empty then [Lemma 35](#) obviously holds true. For any non-empty  $W := \{w_1, w_2, \dots, w_n\} \subset L_\Sigma(\alpha_{\text{abcd}})$  there exist substitutions  $\sigma_i$ ,  $1 \leq i \leq n$ , satisfying  $\sigma_i(\alpha_{\text{abcd}}) = w_i$ . With these  $\sigma_i$  we give the following procedure that constructs a passe-partout  $\beta \in \text{Pat}_{\text{q-tf}, \Sigma}$ :

Initially, we define

$$\beta_0 := \gamma_{1,0} \text{ a } \gamma_{2,0} \gamma_{3,0}^2 \gamma_{4,0}^2 \gamma_{5,0}^2 \gamma_{6,0}^2 \gamma_{7,0}^2 \gamma_{8,0} \text{ b } \gamma_{9,0} \text{ a } \gamma_{2,0} \gamma_{10,0}^2 \gamma_{4,0}^2 \gamma_{5,0}^2 \gamma_{6,0}^2 \gamma_{11,0}^2 \gamma_{8,0} \text{ b } \gamma_{12,0}$$

with  $\gamma_{j,0} := \varepsilon$  for every  $j$ ,  $1 \leq j \leq 12$ .

For every  $w_i \in W$  we define an inverse substitution  $\bar{\sigma}_i : \Sigma^* \longrightarrow X^*$  by

$$\bar{\sigma}_i(A) := \begin{cases} x_{4i-3} & A = a, \\ x_{4i-2} & A = b, \\ x_{4i-1} & A = c, \\ x_{4i} & A = d. \end{cases}$$

For every  $i = 1, 2, 3, \dots, n$  we now consider the following cases:

Case 1: There is no  $A \in \Sigma$  with  $|\sigma_i(x_6)|_A = 1$  and  $|\sigma_i(\text{tf}(\alpha_{\text{abcd}}))|_A = 4$ .

Define  $\gamma_{j,i} := \gamma_{j,i-1} \bar{\sigma}_i(\sigma_i(x_j))$  for every  $j$ ,  $1 \leq j \leq 12$ .

Case 2: There is an  $A \in \Sigma$  with  $|\sigma_i(x_6)|_A = 1$  and  $|\sigma_i(\text{tf}(\alpha_{\text{abcd}}))|_A = 4$ .

Case 2.1:  $A = a$ .

$$\begin{aligned} \text{Define } \gamma_{1,i} &:= \gamma_{1,i-1} \bar{\sigma}_i(\sigma_i(x_1 a x_2 x_3^2 x_4^2 x_5^2)) \bar{\sigma}_i([\sigma_i(x_6^2)/a]), \\ \gamma_{2,i} &:= \gamma_{2,i-1} \bar{\sigma}_i([a \setminus \sigma_i(x_6^2)]), \\ \gamma_{9,i} &:= \gamma_{9,i-1} \bar{\sigma}_i(\sigma_i(x_9 a x_2 x_{10}^2 x_4^2 x_5^2)) \bar{\sigma}_i([\sigma_i(x_6^2)/a]), \\ \gamma_{j,i} &:= \gamma_{j,i-1}, \quad j \in \{3, 4, 5, 6, 10\}, \\ \gamma_{j,i} &:= \gamma_{j,i-1} \bar{\sigma}_i(\sigma_i(x_j)), \quad j \in \{7, 8, 11, 12\}. \end{aligned}$$

Case 2.2:  $A = b$ .

$$\begin{aligned} \text{Define } \gamma_{8,i} &:= \gamma_{8,i-1} \bar{\sigma}(\sigma_i(x_4^2 x_5^2)) \bar{\sigma}_i([\sigma_i(x_6^2)/b]), \\ \gamma_{9,i} &:= \gamma_{9,i-1} \bar{\sigma}_i([b \setminus \sigma_i(x_6^2 x_7^2 x_8 b x_9)]), \\ \gamma_{12,i} &:= \gamma_{12,i-1} \bar{\sigma}_i([b \setminus \sigma_i(x_6^2 x_{11}^2 x_8 b x_{12})]), \\ \gamma_{j,i} &:= \gamma_{j,i-1}, \quad j \in \{4, 5, 6, 7, 11\}, \\ \gamma_{j,i} &:= \gamma_{j,i-1} \bar{\sigma}_i(\sigma_i(x_j)), \quad j \in \{1, 2, 3, 10\}. \end{aligned}$$

Case 2.3:  $A = c$ , and therefore  $\sigma_i(x_4^2 x_5^2) \in \{a, b, d\}^*$ .

Case 2.3.1:  $\sigma_i(x_4^2 x_5^2) \in \{a\}^* \cup \{b\}^* \cup \{d\}^*$ .

$$\begin{aligned} \text{Define } \gamma_{4,i} &:= \gamma_{4,i-1} \bar{\sigma}_i(\sigma_i(x_4 x_5)), \\ \gamma_{5,i} &:= \gamma_{5,i-1} \bar{\sigma}_i(\sigma_i(x_6)), \\ \gamma_{6,i} &:= \gamma_{6,i-1}, \\ \gamma_{j,i} &:= \gamma_{j,i-1} \bar{\sigma}_i(\sigma_i(x_j)), \quad j \in \{1, 2, 3, 7, 8, 9, 10, 11, 12\}. \end{aligned}$$

Case 2.3.2:  $\sigma_i(x_4^2 x_5^2) \in \{a, d\}^+ \setminus (\{a\}^+ \cup \{d\}^+)$ .

$$\begin{aligned} \text{Define } \gamma_{1,i} &:= \gamma_{1,i-1} \bar{\sigma}_i(\sigma_i(x_1 a x_2 x_3^2)) \bar{\sigma}_i([\sigma_i(x_4^2 x_5^2)/a]), \\ \gamma_{2,i} &:= \gamma_{2,i-1} \bar{\sigma}_i([a \setminus \sigma_i(x_4^2 x_5^2 x_6^2)]), \\ \gamma_{9,i} &:= \gamma_{9,i-1} \bar{\sigma}_i(\sigma_i(x_9 a x_2 x_{10}^2)) \bar{\sigma}_i([\sigma_i(x_4^2 x_5^2)/a]), \\ \gamma_{j,i} &:= \gamma_{j,i-1}, \quad j \in \{3, 4, 5, 6, 10\}, \\ \gamma_{j,i} &:= \gamma_{j,i-1} \bar{\sigma}_i(\sigma_i(x_j)), \quad j \in \{7, 8, 11, 12\}. \end{aligned}$$

Case 2.3.3:  $\sigma_i(x_4^2 x_5^2) \in \{a, b, d\}^+ \setminus (\{a\}^+ \cup \{b\}^+ \cup \{d\}^+ \cup \{a, d\}^+)$ .

$$\begin{aligned} \text{Define } \gamma_{8,i} &:= \gamma_{8,i-1} \bar{\sigma}([\sigma_i(x_4^2 x_5^2)/b]), \\ \gamma_{9,i} &:= \gamma_{9,i-1} \bar{\sigma}_i([b \setminus \sigma_i(x_4^2 x_5^2 x_6^2 x_7^2 x_8 b x_9)]), \\ \gamma_{12,i} &:= \gamma_{12,i-1} \bar{\sigma}_i([b \setminus \sigma_i(x_4^2 x_5^2 x_6^2 x_{11}^2 x_8 b x_{12})]), \\ \gamma_{j,i} &:= \gamma_{j,i-1}, \quad j \in \{4, 5, 6, 7, 11\}, \\ \gamma_{j,i} &:= \gamma_{j,i-1} \bar{\sigma}_i(\sigma_i(x_j)), \quad j \in \{1, 2, 3, 10\}. \end{aligned}$$

Case 2.4:  $A = d$ , and therefore  $\sigma_i(x_4^2 x_5^2) \in \{a, b, c\}^*$ .

If, in the conditions of Cases 2.3.1, 2.3.2 and 2.3.3,  $d$  is replaced by  $c$  then the subcases and definitions of Case 2.3 exactly corresponds to what is appropriate for Case 2.4.

Finally, define

$$\beta_i := \gamma_{1,i} a \gamma_{2,i} \gamma_{3,i}^2 \gamma_{4,i}^2 \gamma_{5,i}^2 \gamma_{6,i}^2 \gamma_{7,i}^2 \gamma_{8,i} b \gamma_{9,i} a \gamma_{10,i} \gamma_{11,i}^2 \gamma_{12,i}^2 \gamma_{13,i}^2 \gamma_{14,i}^2 \gamma_{15,i}^2 \gamma_{16,i}^2 \gamma_{17,i}^2 \gamma_{18,i} b \gamma_{19,i}.$$

When this has been accomplished for every  $i$ ,  $1 \leq i \leq n$ , then define  $\beta := \beta_n$ .

For the proof that  $\beta$  indeed is a passe-partout for  $\alpha_{abcd}$  and  $W$ , see the proof of Lemma 34, mutatis mutandis.  $\square$

Thus, we can conclude that, for an alphabet  $\Sigma$  with three or four letters, the class of quasi-terminal-free E-pattern languages is not learnable:

**Theorem 36.** *Let  $\Sigma$  be an alphabet,  $|\Sigma| \in \{3, 4\}$ . Then  $\text{ePAT}_{q\text{-tf}, \Sigma}$  is not inferrable from positive data.*

**Proof.** By Lemmas 34 and 35 and the definition of a passe-partout, there exists a pattern  $\alpha \in \text{Pat}_{q\text{-tf}, \Sigma}$  such that no finite  $W \subseteq L_{\Sigma}(\alpha)$  is a telltale for  $L_{\Sigma}(\alpha)$  with respect to  $\text{ePAT}_{q\text{-tf}, \Sigma}$ . By Theorem 3, this proves Theorem 36.  $\square$

Theorem 28 follows immediately from Theorem 36 and the fact that  $\text{ePAT}_{\Sigma} \supseteq \text{ePAT}_{q\text{-tf}, \Sigma}$ .

## 5.2. Some remarks on the proof

Clearly, the procedures in the proofs for [Lemmas 34](#) and [35](#) implement only one out of many possibilities of constructing the passe-partouts. The definition of the  $\gamma_{j,i}$  in Case 2.3.1 in the proof of [Lemma 35](#), for instance, could be separated in Cases 2.3.1.1 and 2.3.1.2 depending on the question whether or not  $\sigma_i(x_4^2 x_5^2) \in \{a\}^+$ . If so then Case 2.3.1.1 could equal Case 2.3.2, possibly leading to a different passe-partout. It can be easily seen that there are numerous other options like this. On the other hand, there are infinitely many different succinct patterns that can act as a substitute for  $\alpha_{abc}$  and  $\alpha_{abcd}$  in the respective lemmata. Some of these patterns, for instance, can be constructed replacing in  $\alpha_{abc}$  and  $\alpha_{abcd}$  the substring  $\alpha_0 = x_4^2 x_5^2 x_6^2$  by any  $\alpha'_0 = x_p^2 x_{p+1}^2 \dots x_{p+q}^2$ ,  $p > \max\{j \mid x_j \in \text{var}(\alpha_{abcd})\}$ ,  $q \geq 4$ . Hence, the phenomenon described in [Lemmas 34](#) and [35](#) is fairly common in ePAT. Therefore we give some brief considerations concerning the question for the shortest patterns generating a language without telltale with respect to ePAT. Obviously, even for the proof concept of [Lemmas 34](#) and [35](#), shorter patterns are suitable. In  $\alpha_{abc}$ , e.g., the substring  $x_3^2$  and the separating terminal symbol  $a$  in the middle of the pattern can be removed without loss of applicability; for  $\alpha_{abcd}$ , e.g., the substrings  $x_3^2$  and  $x_7^2$  can be mentioned. Nevertheless, we consider both patterns in the given shape easier to grasp, and, moreover, we expect that the indicated steps for shortening  $\alpha_{abc}$  and  $\alpha_{abcd}$  lead to patterns with minimum length. More precisely, we define the patterns  $\alpha'_{abc}$  and  $\alpha'_{abcd}$  by

$$\begin{aligned}\alpha'_{abc} &:= x_1 a x_2 x_4^2 x_5^2 x_6^2 x_7 a x_2 x_8^2 x_4^2 x_5^2 x_6^2, \\ \alpha'_{abcd} &:= x_1 a x_2 x_4^2 x_5^2 x_6^2 x_8 b x_9 a x_2 x_{10}^2 x_4^2 x_5^2 x_6^2 x_{11}^2 x_8 b x_{12}.\end{aligned}$$

Then we conjecture that

- for  $\Sigma_1 := \{a, b, c\}$ ,  $L_{\Sigma_1}(\alpha'_{abc})$  has no telltale with respect to  $\text{ePAT}_{\Sigma_1}$ ,
- for  $\Sigma_2 := \{a, b, c, d\}$ ,  $L_{\Sigma_2}(\alpha'_{abcd})$  has no telltale with respect to  $\text{ePAT}_{\Sigma_2}$  and
- there do not exist any shorter patterns in this property.

Finally, we wish to briefly discuss the extensibility of the proof method in [Section 5.1](#) to larger alphabets. We do not see any straightforward method to extend our way of composing example patterns to  $|\Sigma| \geq 5$  and, in fact, we conjecture the opposite of [Theorem 36](#) to be true for the said alphabet sizes:

**Conjecture 37.** *Let  $\Sigma$  be an alphabet,  $|\Sigma| \geq 5$ . Then  $\text{ePAT}_{\text{q-tf}, \Sigma}$  is inferrable from positive data.*

For our way of reasoning on [Theorem 28](#), the option to exclusively deal with quasi-terminal-free patterns is vital since the decidability of the inclusion problem (as given for appropriate subclasses of  $\text{ePAT}_{\text{q-tf}, \Sigma}$ ; see [Theorem 32](#)) significantly facilitates any considerations on telltales and passe-partouts. Hence, if [Conjecture 37](#) is correct and, still,  $\text{ePAT}_{\text{tf}}$  is not learnable for alphabets with five or more letters then the necessary argumentation might be extremely difficult. On the other hand, if  $\text{ePAT}_{\text{tf}}$  is inferrable for larger alphabets then we anticipate that the corresponding reasoning could provide insights into combinatorics on terminal-preserving morphisms that should also allow us to answer the unresolved equivalence problem for E-pattern languages. For additional information on the latter subject and its connections to the ambiguity of terminal-preserving morphisms, see [Reidenbach \[28,26\]](#).

## 6. Conclusion

In the present paper we have examined the inferrability of E-pattern languages from a combinatorial point of view. In [Section 3](#) we have given two characteristic criteria on the subject: The first main theorem has determined the shortest generators of terminal-free E-pattern languages, and it has demonstrated that these patterns correspond to those strings which are not a fixed point of a nontrivial morphism. The second main theorem has shown that inductive inference of the full class of these languages is equivalent to a combinatorial problem on the ambiguity of morphisms in word monoids. Using these tools, we have proven in [Section 4](#) that the  $\text{ePAT}_{\text{tf}, \Sigma}$  is inferrable from positive data provided that  $|\Sigma| \geq 3$ . Hence, referring to the negative result on  $|\Sigma| = 2$  presented by [Reidenbach \[30\]](#), the learnability of that class is discontinuous with respect to the alphabet size. We have explained that this counter-intuitive phenomenon is caused by differences in the ambiguity of particular substitutions (i.e. morphisms) over binary and ternary alphabets. [Section 5](#) has demonstrated the second discontinuity in the learnability of E-pattern languages: the positive result on terminal-free E-pattern languages cannot be extended to the class of general E-pattern languages

if  $|\Sigma| \in \{3, 4\}$ . The corresponding proof is based on the fact that terminal-preserving morphisms cause types of ambiguity which differ from those of standard morphisms. The case  $|\Sigma| \geq 5$  has been left open.

Our combinatorial methodology has yielded several insights of intrinsic interest into the topology of classes of E-pattern languages and into combinatorics on words and morphisms. In particular, we have shown that, in a combinatorial view and unlike a coding theoretical (i.e. algebraic) context, the properties of morphisms over binary alphabets remarkably differ from those over ternary alphabets. Additionally, we have pointed out that a deeper understanding of general E-pattern languages requires further examinations of the special properties (i.e., in particular, the special ambiguity) of terminal-preserving morphisms.

## For further reading

Refs. [27,29].

## References

- [1] D. Angluin, Finding patterns common to a set of strings, *Journal of Computer and System Sciences* 21 (1980) 46–62.
- [2] D. Angluin, Inductive inference of formal languages from positive data, *Information and Control* 45 (1980) 117–135.
- [3] G.R. Baliga, J. Case, S. Jain, The synthesis of language learners, *Information and Computation* 152 (1999) 16–43.
- [4] C. Choffrut, J. Karhumäki, Combinatorics of words, in: G. Rozenberg, A. Salomaa (Eds.), in: *Handbook of Formal Languages*, vol. 1, Springer, 1997, pp. 329–438 (Chapter 6).
- [5] A. Ehrenfeucht, G. Rozenberg, Finding a homomorphism between two words is NP-complete, *Information Processing Letters* 9 (1979) 86–88.
- [6] G. Filè, The relation of two patterns with comparable language, in: *Proc. 5th Annual Symposium on Theoretical Aspects of Computer Science, STACS 1988*, in: *Lecture Notes in Computer Science*, vol. 294, 1988, pp. 184–192.
- [7] D.D. Freydenberger, D. Reidenbach, J.C. Schneider, Unambiguous morphic images of strings, *International Journal of Foundations of Computer Science* 17 (2006) 601–628.
- [8] E.M. Gold, Language identification in the limit, *Information and Control* 10 (1967) 447–474.
- [9] D. Hamm, J. Shallit, Characterization of finite and one-sided infinite fixed points of morphisms on free monoids, Technical Report CS-99-17, Dep. of Computer Science, University of Waterloo, 1999. <http://www.cs.uwaterloo.ca/shallit/papers.html>.
- [10] T. Harju, J. Karhumäki, Morphisms, in: G. Rozenberg, A. Salomaa (Eds.), *Handbook of Formal Languages*, vol. 1, Springer, 1997, pp. 439–510 (Chapter 7).
- [11] T. Head, Fixed languages and the adult languages of OL schemes, *International Journal of Computer Mathematics* 10 (1981) 103–107.
- [12] S. Jain, D. Osherson, J.S. Royer, A. Sharma, *Systems That Learn*, second edition, MIT Press, Cambridge, MA, 1999.
- [13] T. Jiang, E. Kinber, A. Salomaa, K. Salomaa, S. Yu, Pattern languages with and without erasing, *International Journal of Computer Mathematics* 50 (1994) 147–163.
- [14] T. Jiang, A. Salomaa, K. Salomaa, S. Yu, Decision problems for patterns, *Journal of Computer and System Sciences* 50 (1995) 53–63.
- [15] S. Lange, *Algorithmic Learning of Recursive Languages*, Mensch & Buch, Berlin, 2000.
- [16] S. Lange, R. Wiehagen, Polynomial-time inference of arbitrary pattern languages, *New Generation Computing* 8 (1991) 361–370.
- [17] S. Lange, S. Zilles, Formal language identification: Query learning vs. Gold-style learning, *Information Processing Letters* 91 (2004) 285–292.
- [18] S. Lange, S. Zilles, Relations between Gold-style learning and query learning, *Information and Computation* 203 (2005) 211–237.
- [19] M. Lipponen, G. Păun, Strongly prime PCP words, *Discrete Applied Mathematics* 63 (1995) 193–197.
- [20] M. Lothaire, *Combinatorics on Words*, Addison-Wesley, Reading, MA, 1983.
- [21] G.S. Makanin, The problem of solvability of equations in a free semi-group, *Soviet Mathematics Doklady* 18 (1977) 330–334.
- [22] A. Mateescu, A. Salomaa, Finite degrees of ambiguity in pattern languages, *RAIRO Informatique théorique et Applications* 28 (1994) 233–253.
- [23] A. Salomaa, Patterns, in: G. Rozenberg, A. Salomaa (Eds.), in: *Handbook of Formal Languages*, vol. 1, Springer, 1997, pp. 230–242 (Chapter 4.6).
- [24] A.R. Mitchell, Learnability of a subclass of extended pattern languages, in: *Proc. 11th Annual Conference on Computational Learning Theory, COLT 1998*, 1998, pp. 64–71.
- [25] E. Ohlebusch, E. Ukkonen, On the equivalence problem for E-pattern languages, *Theoretical Computer Science* 186 (1997) 231–248.
- [26] D. Reidenbach, An examination of Ohlebusch and Ukkonen’s conjecture on the equivalence problem for E-pattern languages, *Journal of Automata, Languages and Combinatorics* 12 (3) (2007).
- [27] D. Reidenbach, A discontinuity in pattern inference, in: *Proc. 21st Annual Symposium on Theoretical Aspects of Computer Science, STACS 2004*, in: *Lecture Notes in Computer Science*, vol. 2996, 2004, pp. 129–140.
- [28] D. Reidenbach, On the equivalence problem for E-pattern languages over small alphabets, in: *Proc. 8th International Conference on Developments in Language Theory, DLT 2004*, in: *Lecture Notes in Computer Science*, vol. 3340, 2004, pp. 368–380.
- [29] D. Reidenbach, On the learnability of E-pattern languages over small alphabets, in: *Proc. 17th Annual Conference on Learning Theory, COLT 2004*, in: *Lecture Notes in Artificial Intelligence*, vol. 3120, 2004, pp. 140–154.
- [30] D. Reidenbach, A non-learnable class of E-pattern languages, *Theoretical Computer Science* 350 (2006) 91–102.
- [31] R. Reischuk, T. Zeugmann, An average-case optimal one-variable pattern language learner, *Journal of Computer and System Sciences* 60 (2000) 302–335.

- [32] H. Rogers, *Theory of Recursive Functions and Effective Computability*, MIT Press, Cambridge, MA, 1992, 3rd print.
- [33] P. Rossmanith, T. Zeugmann, Stochastic finite learning of the pattern languages, *Machine Learning* 44 (2001) 67–91.
- [34] G. Rozenberg, A. Salomaa, *Handbook of Formal Languages*, vol. 1, Springer, Berlin, 1997.
- [35] K. Salomaa, Patterns, in: C. Martin-Vide, V. Mitrană, G. Păun (Eds.), *Formal Languages and Applications*, in: *Studies in Fuzziness and Soft Computing*, vol. 148, Springer, 2004, pp. 367–379.
- [36] T. Shinohara, Polynomial time inference of extended regular pattern languages, in: *Proc. RIMS Symposia on Software Science and Engineering*, Kyoto, in: *Lecture Notes in Computer Science*, vol. 147, 1982, pp. 115–127.
- [37] T. Shinohara, S. Arikawa, Pattern inference, in: K.P. Jantke, S. Lange (Eds.), *Algorithmic Learning for Knowledge-Based Systems*, GOSLER Final Report, in: *Lecture Notes in Artificial Intelligence*, vol. 961, Springer, Berlin, 1995, pp. 259–291.
- [38] A. Thue, Über unendliche Zeichenreihen, *Kra. Vidensk. Selsk. Skrifter. I Mat. Nat. Kl.* 7 (1906).
- [39] R. Wiehagen, T. Zeugmann, Ignoring data may be the only way to learn efficiently, *Journal of Experimental and Theoretical Artificial Intelligence* 6 (1994) 131–144.
- [40] K. Wright, Identification of unions of languages drawn from an identifiable class, in: *Proc. 2nd Annual Workshop on Computational Learning Theory*, COLT 1989, 1989, pp. 328–333.
- [41] T. Zeugmann, S. Lange, A guided tour across the boundaries of learning recursive languages, in: K.P. Jantke, S. Lange (Eds.), *Algorithmic Learning for Knowledge-Based Systems*, GOSLER Final Report, in: *Lecture Notes in Artificial Intelligence*, vol. 961, Springer, Berlin, 1995, pp. 190–258.